how to initialize Commodore 64 disk

```
OPEN 15,8,15,"N0:DISK,EA"
```

$$\begin{array}{r} {}^{4}\cancel{5}15 \\ \underline{39} \\ 2;10 \end{array}$$

To be done: 9.13

- disable RESTORE key → too easy to hit instead of RETURN
- input trap for worm names
- outline territories at end of game?
- erase previous screens

KEYS

F1	warm boot – takes you back to splash screen
F3	takes you to next worm
F5	changes type of worm
F7	start
RUN/STOP	pause toggle
0–9	speeds
S	save worm
U	update worm
L	load worm

DIRECTORY

9:45 - 12:59

WORMS? CC~~#~~K

Opening

Electronic Arts should be one color

a G# 0 is left from the previous screen

Game

When a worm dies its **trail** and title/score should light up.

Saving/Directory

Saving works

★ When trying to put more than 4 letters in the title of the worm the ~~#~~ score line scrolls ~~the line with the scores.~~ to the left, off the screen and reappears from ~~left~~ right side of screen and disappears behind FILE NAME: ---

The Directory command doesn't work at all.

when D is pressed it prints Directory and dosen't erase the rest of the line ~~it's on~~. from the previous message

★ RESTORE starts game again but does not erase previous screen

★

FILE NAME: WWWWK disappears ←

← 2 1 4 20

SAME QQQQ TWO FOUR

5.75
4.00
5.50
7.00
22.25

10:00-11:35, 3:35-4:40.

400
WORMS
FRI

Play Testing

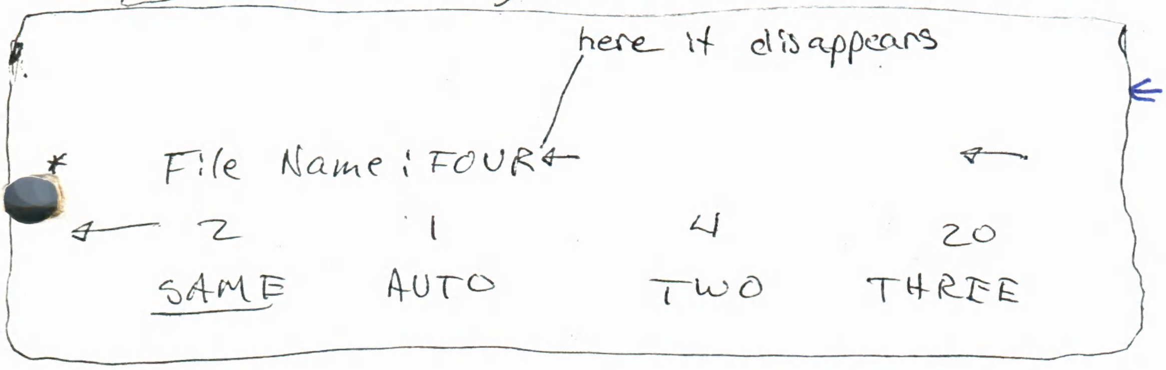
Worms? CG4K

Opening
✓Electronic Arts should be one color
✓a G# 0 is left from the previous screen

Play
When a worm is killed its trail and title/score don't light up making it hard to tell who died.
The sound when a worm dies is to short.
~~The~~ entire game seems to slow

Save / Directory
* When trying ~~hitting~~ to put more than four letters in the File name of the worm the score line scrolls to the left, off the screen and reappears from the right side of the line above and then disappears behind the file name.

the Directory command doesn't work.
when D is pressed it prints Directory, not Directory? <RETURN> and doesn't erase the rest of the line from the previous message.
RESTORE starts game over but dosen't erase previous screen.



← • input trap (backspace, too)
• DIRECTORY
~~• Joysticks~~ don't do

Play Testing

Worms? C64K

Opening

Electronic Arts should be one color

a G# 0 is left from the previous screen

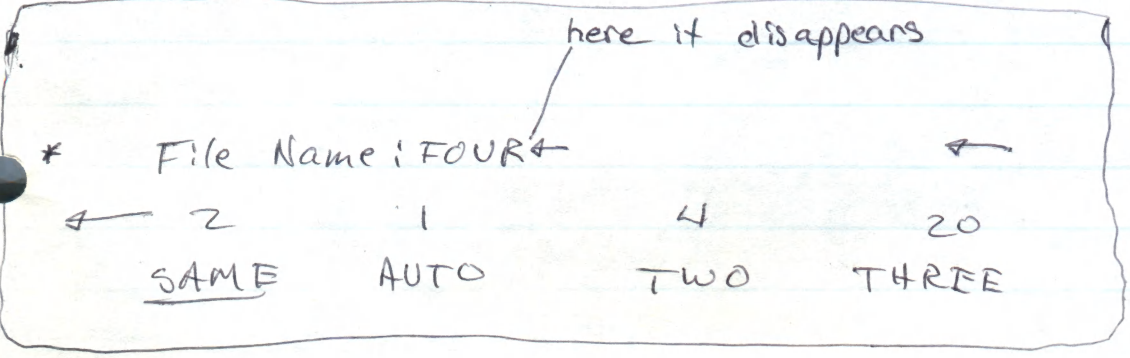
Play

When a worm is killed its trail and title/score don't light up making it hard to tell who died.

Save / Directory

* When ~~trieing~~ trying to put more than four letters in the File name of the worm the score line scrolls to the left, off the screen and reappears from the right side of the line above and then disappears behind the file name.

the Directory command doesn't work.
When D is pressed it prints Directory, not Directory?
<RETURN> and doesn't erase the rest of the line from the previous message.
RESTORE starts game over but dosen't erase previous screen.



To Save Worms:

First:
- Load the Worms? program disk, remove the program disk from the drive and insert the blank, formatted disk (using Atari DOS) into the drive. Remember: If you make a mistake, you can press any key to go back to the SELECT/OPTION/START menu.

Then:
- Use SELECT to indicate which worm you want to save.
- Press $\boxed{S}$. The computer will respond SAVE? RETURN.
- Press RETURN key for "yes". Or else if you don't want to save that worm, press any other key to return to the SELECT/OPTION/START menu.
- The computer will ask FILE NAME: Enter a worm name that is no more than 4 characters long. Then press RETURN. The screen will go blank as the worm is saved onto the disk.

To Load Worms:

First:
- Load the WORMS? disk. Then put the disk with the saved worms in the disk drive.

Then:
- Use the SELECT key to indicate which color you want your saved worm to play.
- Press $\boxed{L}$. The computer will respond LOAD? RETURN.
- Press RETURN and the computer will ask FILE NAME:
- Enter the worm's name and press RETURN. The screen will go blank as the worm is loaded.

To Update Worms (for teaching a "saved" worm new moves):

First:
- Put the disk with the saved worms in the disk drive.

Then:
- Press the SELECT key to choose which worm is being updated.
- Press $\boxed{U}$ for update. The computer will respond UPDATE (worm's name)? RETURN
- Press RETURN and the new moves will be added to the worm's file.
- The screen will go blank during this time.

Directory:

First:
- Put your disk with the saved worms in the disk drive.

Then:
- Press $\boxed{D}$. The computer will respond DIRECTORY? RETURN.
- Press RETURN and the names of your saved worms will be listed at the top of the screen.

When you are finished, just press the function key to begin your next game.

Additional Credits:

Target Compiler: Nautilus Systems
Forth System: Norm Lane
Charles Moore and whole fig-Forth community

WORMS?
COMMAND SUMMARY
Atari 400/800/1200XL

Number of Players - One to four.

To Load - Type LOAD"EA", 8, 1 and hit RETURN ~~See your Atari Owner's Guide for instructions.~~

To Select Worms:

F ~~SELECT~~ key - For moving to the next worm position.

F ~~OPTION~~ key - For choosing a type of worm (NEW, WILD, etc.) for each position.

To Start - F7 ~~START~~ key.

NOTE: Use F ~~SELECT~~ and F ~~OPTION~~ only if you want to change worm types. Otherwise just use START to begin another game.

Play Commands	Using Keyboard	Using Joysticks ~~Paddles~~
Choose worm direction	SPACE BAR	Turn ~~paddle~~ joystick
Move	Press any key	Red fire button

Keyboard movements and joystick ~~paddle~~ movements can be used in the same game.

~~Note: Paddles in Port #1 control the gold and pink worms; in Port #2, the blue and green worms.~~

Susan

Wednesday 4.13

Trip —

I'm getting together with David Maynard on Friday to determine the C64 milestones for WORMS?. Before the milestone dates can be figured, David needs to know whether the product will be on disk, cassette, or both. The SAVE-WORMS code is imbedded throughout the program, and the assumption is that cassette versions would not have the save-worm feature.

Is there a possibility that the Exec Committee could make a determination of which medium we will support for the C64 tomorrow (4.14)? Can you assume Disk for now with the possibility of changing at the "last" minute to cassette? If not, assume disk anyway... what the hell...

Thanks,

Susan

P.S. I ~~[illegible]~~ really appreciate your feedback on the Trost product. Could we set up an appt. to go over some of the "see me" ideas?

Yes arrange with Jill

WORMS? tm

COMMAND SUMMARY

Commodore 64

Number of Players: One to four

To Load: Turn on the disk drive, turn on the computer, and insert the program disk. Type LOAD"EA",8,1 followed by a RETURN. (See your Disk Drive User's Manual, if you need additional instructions.

Play Commands

Starting:	F1 key	Resets the program
	F3 key	Lets you choose a type of worm (NEW, WILD, etc.)
	F5 key	Lets you select the next worm position (red, green, etc.)
	F7 key	Starts the game; during a game, it takes you back to the F3/F5/F7 menu
Moving:	SPACE BAR	Changes the worm's direction (when the worm is waiting to be told what to do)
	any letter key	Tells the worm to move
Other:	RUN/STOP key	Pauses/restarts game
	0 to 9 keys	Sets the speed of the worms (0 is the slowest, 9 is fastest, 6 is the default)

NOTE: Use F3 and F5 only if you want to change worms. Otherwise, just hit F7 to start another game.

YOUR GAME MANUAL IS LOCATED INSIDE THE FRONT PANEL OF THIS PACKAGE.

Note: Ignore the commands printed on the fourth page of the manual. The Commodore 64 version of WORMS? does not allow territories or worms to flash.

Commodore 64 is a registered trademark of Commodore Business Machines, Inc.

To Save Worms:

First: *Initialize a blank disk, following the instructions in your Disk Drive User's Manual.

Then: *Load the WORMS? program disk, remove the program disk from the drive, then insert a blank, formatted disk into the drive.

Remember: If you make a mistake, you can press any key to go back to the F3/F5/F7 menu

*Use F5 to indicate which worm you want to save.

*Press S . The computer will respond SAVE? RETURN.

*Press RETURN key for "yes." If you don't want to save that worm, press any other key to return to the F3/F5/F7 menu.

*The computer will ask FILE NAME: Enter a worm name that is no longer than 4 characters long. Then press RETURN . The screen will go blank as the worm is being saved onto the disk.

To Load Worms:

First: *Load the WORMS? disk. Then put the disk with the saved worms into the disk drive.

Then: *Use the F5 key to indicate which color you want your saved worm to play.

*Press L . The computer will respond LOAD? RETURN.

*Press RETURN and the computer will ask FILE NAME:

*Enter the worm's name and press RETURN . The screen will go blank as the worm is loaded.

To Update Worms (for teaching an already "saved" worm new moves):

First: *Put the disk with the saved worms in the disk drive.

Then: *Press the F5 key to choose which worm is being updated.

*Press U for update. The computer will respond UPDATE [worm's name]? RETURN.

*Press RETURN and the new moves will be added to the worm's file. The screen will go blank during this time.

Directory:

First: *Boot the disk with the saved worms on it.

Then: *Type LOAD"$",8 and hit the RETURN key. Then type LIST. You will see the list of saved worms: [worm name].WRM

A COPY OF MAYNARD'S SOLITAIRE (shown on the package) IS INCLUDED ON THE PROGRAM DISK. IT IS CALLED DEMO AND CAN BE LOADED FOLLOWING THE INSTRUCTIONS ABOVE.

Additional Credits:

Target Compiler: Nautilus Systems

Forth System: Norm Lane

Charles Moore and the whole fig-FORTH community

WORMS - C64
[illegible]

PRELIMINARY BILL OF MATERIALS

```
MFBOMRPT V 0.1    B I L L   O F   M A T E R I A L S   R E P O R T        PAGE:
------------------------------------------------------------------------------------


                 QTY
 DP#     COMP #  ASMBLY AVER-COST  DLR-PRICE  DESCRIPTION
------  ------  ------ ---------- ---------- --------------------------------

001022                      .00       21.00
         100501    1        .32         .32  ALBUM, WORMS
         100505    1        .20         .20  MANUAL, WORMS
         102202    1        .04         .04  DISK LABEL, WORMS-C64
         102203    1        .04         .04  FIVE CARD, WORMS-C64
         102204    1        .02         .02  SKU STKR, WORMS-C64
         790101    1        .04         .04  DISK SLEEVE, W/ E.A. LOGO
         790102    0        .00         .00  DISK, SS/SD, NO NOTCH, PRO S1
                       ---------- ----------
                            .66         .66
```

APPROVED BY: ____________________ DATE: 9/1/83

Susan -
a fan letter!
Jill
(Trip will be sending
a "thank you")

RECEIVED
JUN 24 1983

Michael Moore
[redacted]
San Francisco, CA 94117

Electronic Arts
2755 Campus Ave.
San Mateo, CA 94403

Dear Electronic Arts,

I've never written a company simply to praise a product, but having recently bought "Worms?" I must say that it is the most imaginative and fun game I've played in a while. Actually, I haven't bought many games in the past months because most are so repetitive and boring.

I've had my Atari for about two years, and have played quite a few games on it, but recently had despaired of doing anything entertaining on it. Many of the games are so lacking in imagination or innovation that I was beginning to feel a sense of deja vu at the local store. Worms? however, was instantly appealing and I bought it on the spot (something I rarely do).

Anyway, instead of rambling on, I just want to say, "keep up the good work, and may your imaginations hold up". Your games are a credit to the intelligence of the users, and I hope they prove that there's a market for games that might not appeal only to adolescents.

Yours truly,
Michael Moore
Michael Moore

black

ORMS?

BUGS

- can backspace over prompts
- F key toggle or move?
  # keys speed or move?

ELECTRONIC ARTS
2755 CAMPUS DRIVE
SAN MATEO, CA 94403
TEL 415 571-7171

David—

We found out that the Source Code you sent was not the right stuff. What I actually received that night were two disks of object code and one disk of source code. We get to do it again!!

Here is precisely what is needed:

1. annotated, commented source screens including kernel

2. listings of #1

3. executable object code

4. supplementary documentation for special code

Susan

Jeff
Rich

To keep w/
product descrip.
for Sales
mtg.
Susan

WORMS?

By David S. Maynard

WORMS? is an art and strategy game for one-to-four players. Brightly colored worms quickly move from one dot to another on the black matrix-like background. The worms create music as they move and leave a colored trail behind. Each dot can have six possible trails going in an out of it. Points are gained when all six trails have been generated from a dot. Players train their worms where to move, and they will remember - moving automatically until they reach a new decision point. Players can compete against different computer opponents or other human-type players. [illegible]

Suggested description of WORMS? for Regis McKenna Product Descriptions

Regis McKenna, Inc.
Marketing & Communications Strategy

ELECTRONIC ARTS PRODUCT DESCRIPTIONS

Electronic Arts Statement of Purpose

ARCHON

By FreeFall

ARCHON is a combination strategy and fantasy game for up to two players. It pits a light team against a dark team on a chess-like board. The object is to capture five power points on the board by moving the various pieces strategically as well as by winning victories on the battlefield which appears when individual pieces are matched up.

Atari version available in May

Cost:

PINBALL CONSTRUCTION SET

By Bill Budge

PINBALL CONSTRUCTION SET is a computerized erector set. It allows the user to be creative by designing his own pinball game by using a menu of pinball machine parts like bumpers, balls, and paddles. Design options also include a sophisticated graphics paint board, wiring kits, and physical properties (like elasticity and gravity). An instruction manual is included, but is not necessary for basic design.

Atari version available in May, Apple version available from Electronic Arts in July.

Cost:

M.U.L.E.tm

By Ozark Softscape

M.U.L.E.tm is a game for up to four players. It is a simulation game in which players compete for financial advantage on a foreign planet. Play requires both strategy and dexterity. The game allows players to ~~both~~ compete against the computer and also inspires interpersonal bargaining and interaction between the players during the auction phase which concludes each playing turn. There are three levels of play and can be enjoyed by children, teenagers, and adults. (The M.U.L.E. is a "Multiple Use Labor Element.)

An instruction manual accompanies the software which explains the scenario and rules of the game. For the player that is interested in understanding the strategic origins of the game, the manual also includes a basic description of the economic laws on which the play is based.

Atari version available in May.

Cost:

HARD HAT MACK

By

WORMS?

By David S. Maynard

WORMS? is an art and strategy game for up to four players. Played against a black matrix like background, the colored Worms are programmed by players ~~in order~~ to capture ~~sets of six dots. The Worms continue their momentum automatically.~~ the territory surrounding each dot. Territories are captured by ~~drawing~~ moving ~~one of~~ along the six

Atari version available in May.

Cost:

⓪ priorities

3.14.83
3.11.83

David — This is the "hard copy" re worms that I said I would give you.

Package

- ① ✓ I need your list of acknowledgements ASAP
- Your picture will probably be taken 3/17 or so (definite time TBD)
- ✓ I will be asking Martin Gardner for an endorsement

Design Changes

- ② • save worm feature to be added
  - need to be able to save to Worm-only separate disk
  - need to be able to appoint saved worm to particular color/slot
  - ~~Want to~~
- worm teams: be able to play one-on-one and "mixed doubles"
- ③ ✓ install PAUSE mechanism (P) + 'ESC
- ① ✓ underline whose move it is [NEW]
- ② • ~~no~~ gene editor
- no breeding
- ✓ no HAND option
- ✓ • slow down #1 speed even more and calibrate the other speeds accordingly
- ✓ • user interface — make use of ~~joystick~~, paddle or keyboard invisible to user, if possible, so user doesn't have to designate

① ~~• add joystick interface~~

① ~~• add 2 keyboard interfaces:~~ space bar & any other key

W E
A D
Z X

O P
K ;
; :

Manual

✓ • I will have a rough draft done by 3/17 – you will need to check it (Then Bing gets it for rewrite.)

Contest

• We are having a worm-training contest where players can sent in disks (to Bing). They want a one-on-one play and mixed doubles. Bing is handling the judging, etc.

Milestones – These dates are firm, if we want a May 13 ship date

3/21	SW Delivery
3/17	Packing concept approved
4/1	Manual approved (final draft)
3/22	latest date for author pictures

ok?
Susan

yes to Martin Gardner

3-10-83
Worms

Package copy
- directions need to be much more explicit
  too frustrating otherwise
- new interview — wants to add acknowledgments } add to credits
- pictures of David

← get list from David

User Interface
- keyboard option? yes > on stand. screen
  for 2 or 4?
  where?

prompt for illegal moves?
joysticks?
not many paddle owners

- save worms? (user initially feels ripped off) — YES!

Design changes
- no HAND option
?. - no worms at bottom
- new interface (see above) — YES
- underlining ~~p~~ who's move it is
- pause key
- START works anytime (even in middle of game)
- gene editor (?) } probably not
- breeding (?)

worm teams
one-on-one
mixed doubles

~~take 300~~

Trained Worms contest? yes ~ be able to load up to 4 worms in any "slot"
ok w/ Rich

FOCUS ON WORMS 1/20/83

Kids didn't understand it. We have to do something up front to explain what's happening. How much can be described on package?

Action:

1. Understand it myself.
2. See what Andy has planned for the package in the way of instructions.
3. Get info from Maynard on how scoring takes place and what strategies can be used to win. We will need this for both package and "manual."

Tutorial - on line?

Hubs of wheel spokes

Color the points (hubs) according to who won it?

2/3

Gordie
re Martin Gardner

Data comm

letter — letter from him

Dear Pat

This will confirm our telephone

I consent to your use of my name (as follows):

first consent/release

on package
in ads

This game was mentioned
Jan

places where it might be used

May want him
as explicit as possible

Run the copy by him.
(Clause in agreemt
5 days

Can't quote "substantial
amt"

20 words or more
He described it as
"   "

if C#, go for consent
Mag has Ⓒ usually.

Martin Gardner 2/

Refer to ~~fact~~ it was in his column

He says its ok to say — 1st publicized by Martin Gardner

Gosper made 17 min film on Life.

Atari - last 3 chapters

WH Freeman / Sci American

2 colums on life + 3rd new one

Maynard

SIRIUS – He didn't like them
Did not leave a copy
Got a non-disclosure.

WORMS

Egypt -10%
confiscated his demo because no one was getting any work done

39^2
56^2

12/9

Sci Am or MARTIN GARDNER ?
415 Madison Ave
NY 10017

legal :- violation of algorithm?

Promotional opportunity.

won't make or break product

12/7

need terms

mystery - dumb
computer - smart
no change - keeps intelligence

NAMES

---

challenger
champion

Kinetic Art

---

to learn - play against

running score ([illegible]) - at bottom
& final

ADD? - Save a worm
breed a worm : intelligence
Choose : 6 tones for direction
(user could choose)
& tempo
may take longer in 1 dir
than another

Joystick

Maynard 11/16

New splash screen - worms write

Worms squiggle - more wormish

take jitter out of paddles

Speed more constant when many worms

Will show score on bottom

? Running score -
get 1 or 10 for full hex.
and 0 for partial

Now get 6 pts for full hex
✱ 6 lives

now if you have almost hex
& someone else steals it
you get -4
they get +6

seems better

Attract mode - 4 random worms

Apples invites
green strikes

WORMS

Standard for ATARI
{OPTION — type of player
SELECT — chooses player
START —

Player 1 Gold
2
3
4

asleep
new
mystery — CPU player —
no change —?, 1 last game
Racer
→ Racer stops where he doesn't

Screen flashes ← — bright worm if legal
— flashes color whose turn it is

START sometimes resets

SNAILS
ALLIGATOR
SNAKE
SUSHI
SPERMS

Score on Dots
Select wrap or reflect.
$2^6 = 64$ are they color smart? NO

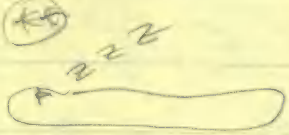
training Mode? — for me
for Joe — strategy

What's average game time?
settable time-outs
decreasing time limits on time out — no good. Game's getting harder, & you're getting less time to think

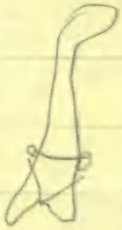
SPLASH Screen

Porfirio

asleep



new



racer

mighty

Scoring

Do something more than show area color — Show NUMERIC score at bottom of screen.

yes

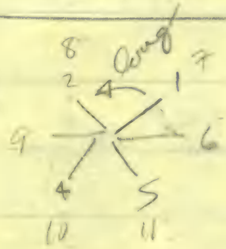
___ ___ ___ ___

2376 30 1820

Paddle

full rt

go CCW

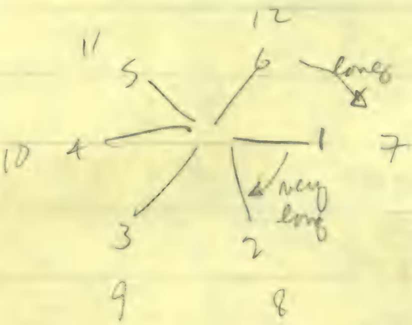


2 times then lots of fiddly till full [rt left] with no action

improved

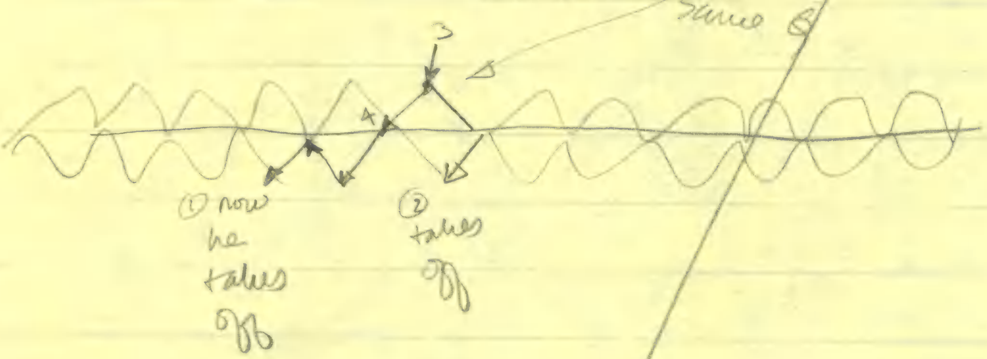
full left

go CW



NEW

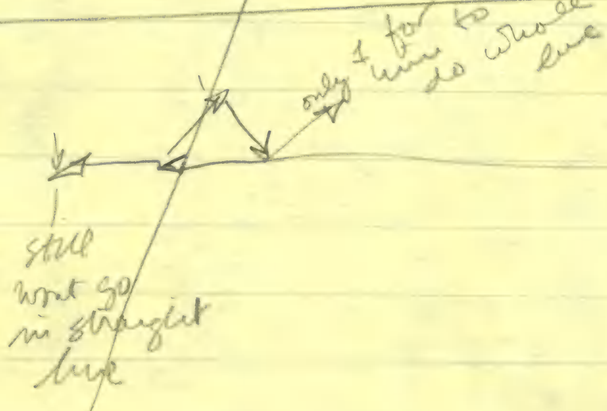
why do I have to do this twice to make him go in the straight line ?



NO CHANGE (= remember)

He does all of the above he did above.

NEW


1
only 1 for him to do whole line
still wont go in straight line

COMMAND • SUMMARY

WORMS?™

ATARI HOME COMPUTERS

Number of Players – One to four.

To Select Worms:

SELECT key – For moving to the next worm position.

OPTION key – For choosing a type of worm (NEW, WILD, etc.) for each position.

F7 To Start – START key.

NOTE: Use SELECT and OPTION only if you want to change worm types. Otherwise just use START to begin another game.

Play Commands	Using Keyboard	Using Paddles
Choose worm direction	SPACE BAR	Turn paddle
Move	Press any key	Red fire button

Keyboard movements and paddle movements can be used in the same game.

NOTE: Paddles in Port #1 control the gold and pink worms; in Port #2, the blue and green worms.

Load Notice:
- This program will not run with BASIC or a cartridge in the system.
- Atari 600XL, 800XL, 1400XL or 1450XL owners must hold down the OPTION key while turning the computer on.
- See your Atari Owner's Guide for detailed instructions.

YOUR GAME MANUAL IS LOCATED INSIDE THE FRONT PANEL OF THIS PACKAGE.

ELECTRONIC ARTS™

To Save Worms:

First:
- Load the Worms? program disk, remove the program disk from the drive and insert the blank, formatted disk (using Atari DOS) into the drive. Remember: If you make a mistake, you can press any key to go back to the SELECT/OPTION/START menu.

Then:
- Use SELECT to indicate which worm you want to save.
- Press S. The computer will respond SAVE? RETURN.
- Press RETURN key for "yes." Or else if you don't want to save that worm, press any other key to return to the SELECT/OPTION/START menu.
- The computer will ask FILE NAME: Enter a worm name that is no more than 4 characters long. Then press RETURN. The screen will go blank as the worm is saved onto the disk.

To Load Worms:

First:
- Load the WORMS? disk. Then put the disk with the saved worms in the disk drive.

Then:
- Use the SELECT key to indicate which color you want your saved worm to play.
- Press L. The computer will respond LOAD? RETURN.
- Press RETURN and the computer will ask FILE NAME:
- Enter the worm's name and press RETURN. The screen will go blank as the worm is loaded.

To Update Worms (for teaching a "saved" worm new moves):

First:
- Put the disk with the saved worms in the disk drive.

Then:
- Press the SELECT key to choose which worm is being updated.
- Press U for update. The computer will respond UPDATE (worm's name)? RETURN.
- Press RETURN and the new moves will be added to the worm's file.
- The screen will go blank during this time.

Directory:

First:
- Put your disk with the saved worms in the disk drive.

Then:
- Press D. The computer will respond DIRECTORY? RETURN.
- Press RETURN and the names of your saved worms will be listed at the top of the screen.

When you are finished, just press the function key to begin your next game.

Additional Credits:

Target Compiler: Nautilus Systems
Forth System: Norm Lane
Charles Moore and whole fig-Forth community

REFERENCE CARD

FOR WORMS?

Atari 400/800/1200

To Begin:

SELECT key — Moves you to the next worm. The SELECT/OPTION/START line is the same color as the selected worm

OPTION key — Allows you to choose the type of worm, you want

START key — Begins the game. In the middle of a game, START takes you back to the SELECT/OPTION/START menu.

Use SELECT and OPTION only if you want to change worm types. Otherwise just use START to begin another game.

To Move Worms:

Keyboard
- Use space bar to choose worm direction
- Press any key to move

Paddles
- Use paddle to choose one of the six directions
- Press red fire button to move

Port #1 is used for paddles controlling the gold and pink worms.

Port #2 is used for paddles controlling the blue and green worms.

Keyboard movements and paddle movements can be used in the same game.

Other:

keys 1 to 9 — Set speed of the worms. Speed can be set any time during the game.

1 is slowest
7 is the default
9 is fastest

ESC key — Halts game; press key again to resume game

G	Turns grid off so dots disappear; press key again to have dots reappear. (This command can only be used at the beginning of a game.)
F	Territories that flash because they have had the last (sixth) trail laid continue flashing for the rest of the game; press this key again to turn continual flashing off. (This command can ~~only~~ be used ~~at~~ any time ~~the beginning of a game~~.)
S	Saves worms onto separate disk
L	Loads worms from separate disk
U	Updates already saved worms
D	Directory of saved worms

(See To Save Worms for full explanation of the last four commands.)

To Save Worms

Preparing

You must format a blank disk, using Atari DOS. When saving and loading worms, first load the game program into the Atari. Then, remove the game program disk from the drive and insert the blank, formatted disk into the drive.

Remember that if you make a mistake, you can press any key to go back to the SELECT/OPTION/START menu.

Saving Worms

- Use the SELECT key to indicate which worm you want to save.
- Press S for save. The computer will respond SAVE? <RETURN>
- Press RETURN key for "yes." If you make a mistake and don't want to save that worm, press any other key to return to the SELECT/OPTION/START menu.

- The computer will ask FILE NAME: Enter a worm name that is no more than

4 characters long. (You might use your initials and numbers.) Then press RETURN.

The screen will go blank as the worm is saved onto the disk.

Loading Worms

The instructions are very similar to Saving Worms. Make sure you have the disk with the saved worms in the disk drive.

- Use the SELECT key to indicate which color you want your saved worm to play.
- Press L for load. The computer will respond LOAD? <RETURN>
- Press RETURN and the computer will ask FILE NAME:
- Enter the worm's name and press RETURN

The screen will go blank as the worm is loaded.

Updating Worms

When you are playing with a saved worm and you are teaching it new moves, you may want the updated moves saved. Again, make sure you have the disk with the saved worms in the disk drive.

- Press the SELECT key to choose which worm is being updated.
- Press U for update. The computer will respond UPDATE (worm's name)? <RET. ~~RETURN~~>
- Press RETURN and the new moves will be added to the worm's file.

The screen will go blank during this time.

Directory

Have your disk with the saved worms in the disk drive.

- Press D for directory. [redacted]. The computer will respond DIRECTORY? <RETURN>
- Press RETURN and the names of your saved worms will be listed at the top of the screen.

When you are finished, just press ~~one of~~ the function key~~s to begin~~ You want to make your Choices for the ~~next~~ game. When the game starts, the playing field will reappear on the screen.

WORMS?

By David S. Maynard

OBJECT: The worms are trained to capture territories by laying trails from dot to dot. Each dot lies in the center of a territory.

SCORING: 1 point goes to the worm that lays the last (sixth) trail in a territory. Scores are shown on the bottom of the screen.

TYPES OF WORMS: There are five worms built into the program. You may also play with worms you have saved from previous games (see To Save Worms). The five worms are as follows:

NEW	The worm is untrained. You must train it as you play the game.
AUTO	This worm is just like NEW, but the computer trains the worm to make smart moves as it plays the game.
WILD	The computer trains the worm randomly for all possible moves before the game begins.
SAME	The worm which played this color in the last game is used again in this game.
-----	The worm is asleep and doesn't play.

STARTING: Choose the type of worm you want for each color that will be playing. Then start the game and begin training your worm! (See the Reference Card for the appropriate commands.)

© 1983 Electronic Arts, San Mateo, CA 94403

HOW TO PLAY WORMS?

(A) The screen is covered with dots. Each dot lies in the center of a territory.

(B) Worms start in the center and can move in one of 6 directions. See the Reference Card for movement commands.

(C) When worms move, they leave a trail behind. Notice a trail is half in one territory and half in another.

(D) Move your worm in the direction you want. NOTE: The worm cannot move along an existing trail. Once a worm recognizes a movement pattern, it will move automatically, until it runs into something (see How Worms Understand Their Training).

(E) The worm that needs a movement command is flashing, and its name is underlined at the bottom of the screen.

(F) When a worm adds a new trail to a dot's territory, all other existing trails in the territory turn to that color.

(G) Several moves later, the red worm is coming back. It is changing the territory's color again.

(H) The red worm has scored a point by laying the last (sixth) trail in the territory. The territory flashes when the point is scored.

HOW TO PLAY WORMS?
p.2

(I) Red and black are headed for the same dot. Once at the dot, any move would be illegal.

(J) The worms die because they cannot move. When all the worms are dead, each conquered territory flashes, the score is shown, and the game is over.

HOW TO CREATE MAYNARD'S SOLITAIRE

You must use only one NEW worm (in any color). Follow the 12 moves shown in red, and you will create Maynard's Solitarre which is shown on the package

WORMS?

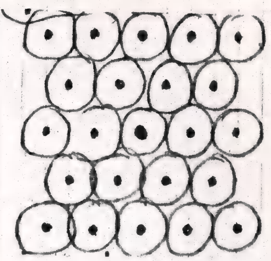
OBJECT:

SCORING:

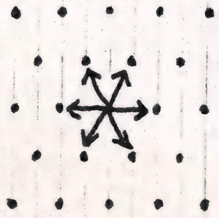
TYPES OF WORMS:

STARTING:

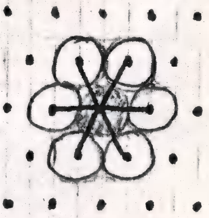
HOW TO PLAY



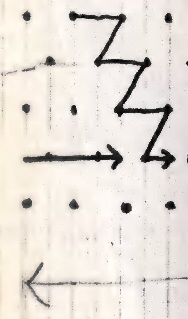
Copy A

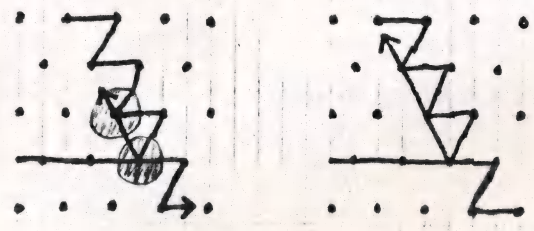


Copy B

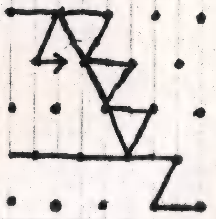


Copy C

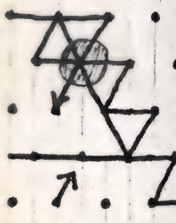




← Copy F →



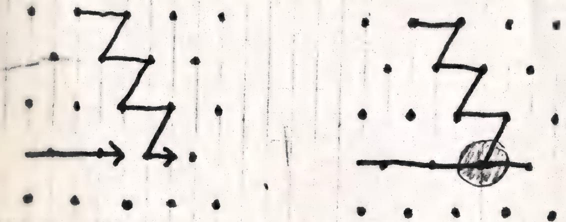
Copy G



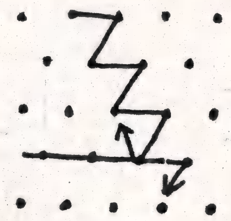
Copy H

TO PLAY WORMS?

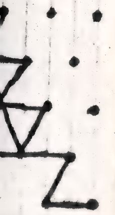
pyC



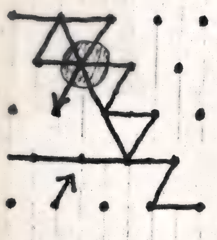
← Copy D →



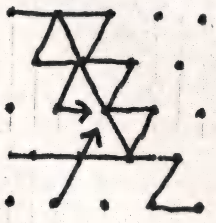
Copy E



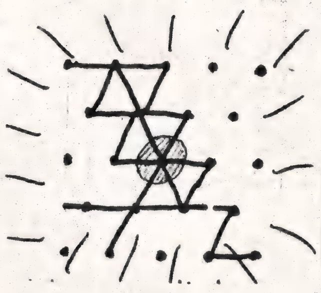
y 6



Copy H

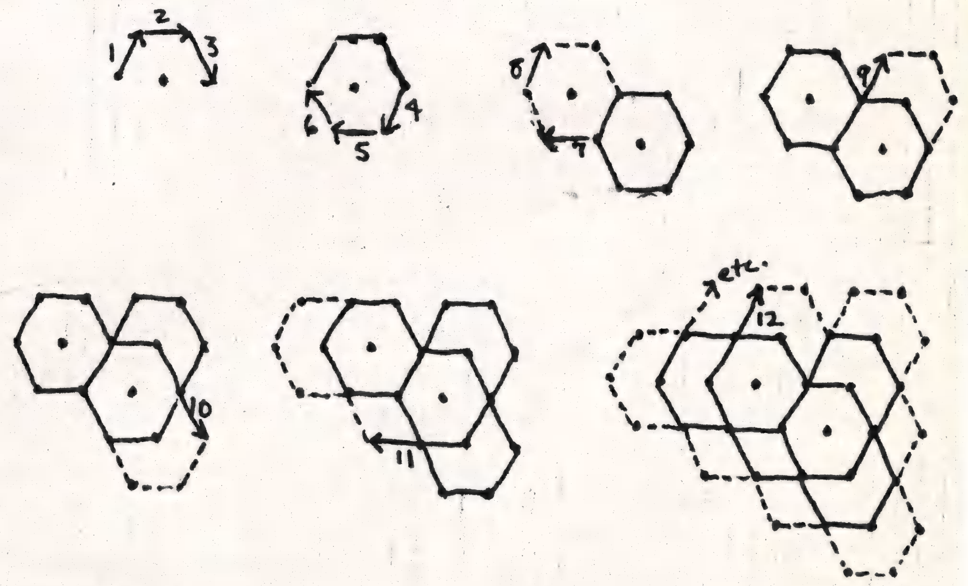


Copy I



Copy J

HOW TO CREATE MAYNARD'S SOLITAIRE



copy

HOW WORMS UNDERSTAND THEIR TRAINING

(AA) ~~In the beginning~~, The worm starts in the center territory. (Pretend the red dot is in the center.) It can move in one of six directions, but it doesn't know which to choose. Let's train the worm to zigzag. Move it to the right.

(BB) Now the worm is in a new territory with one trail going off to the left (─•). Move the worm to the lower left (↙). The worm will now remember that when it encounters ─• it moves ↙ .

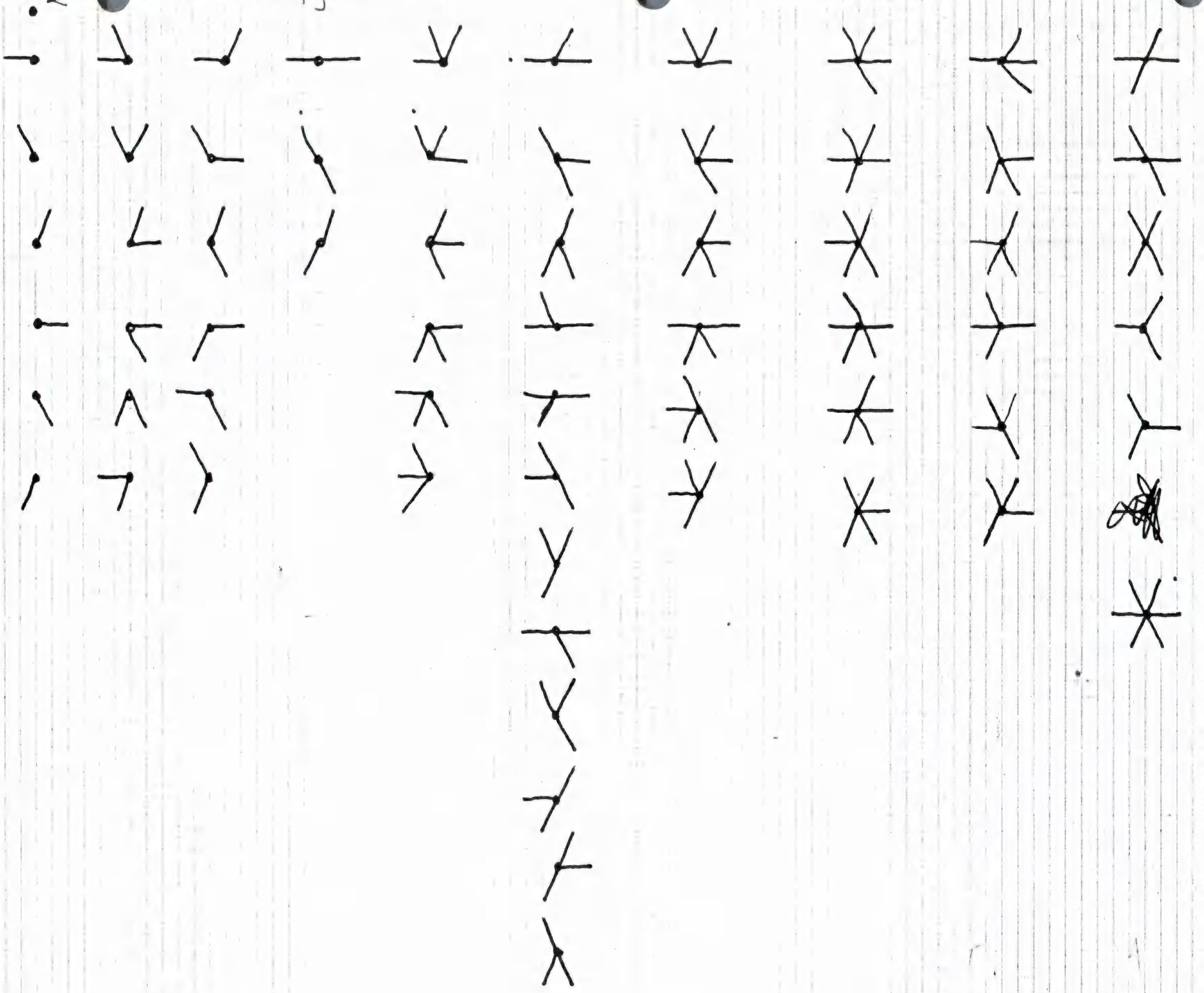
(CC) In this territory, there is a trail coming in from the upper right (•╱). Move the worm to the right (∠→). Now it will remember that when it encounters •╱ , it moves ∠→ .

(DD) Since the worm remembers this configuration from step one, it remembers to move to the lower left. And so it continues.....

(EE) ....until the worm moves into another path or configuration (─∠) it does not recognize, ~~it runs into something. The worm does not recognize the configuration~~. The player must now tell the worm what to do.

(FF) If you move the worm to the lower right, it encounters •╱ in the next territory. The worm recognizes this from Step 3, so it will move ∠→ .

te: This configuration.....



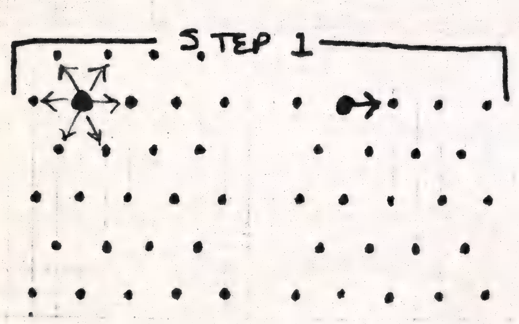
64 possible worm configurations

THE 64 POSSIBLE TRAIL CONFIGURATIONS

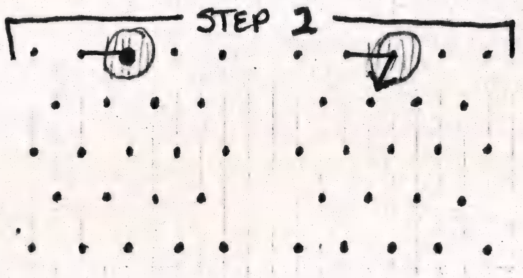
(NOTE: ~~Note that~~ This configuration only appears during the very first move.)

Not all these configurations will show up in the same game.

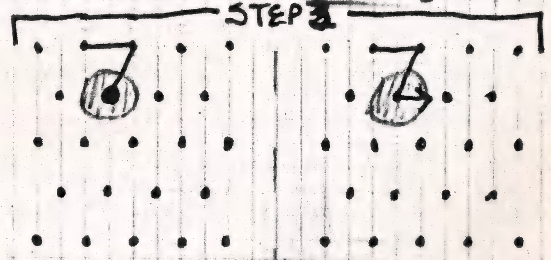
HOW WORMS UNDERSTAND THEIR TR



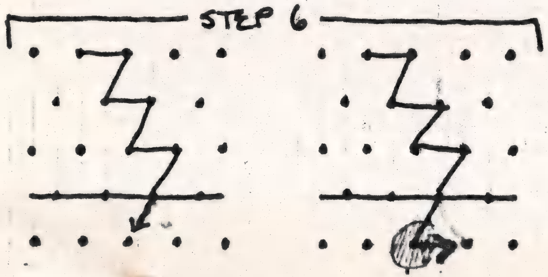
copy AA



copy BB



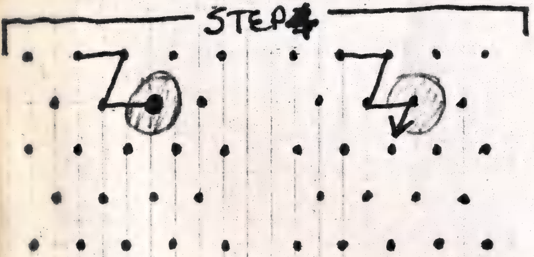
copy CC



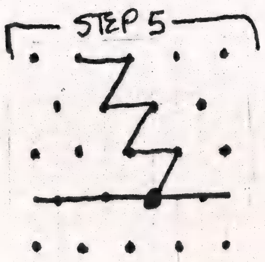
copy FF

64 configura on

D THEIR TRAINING



copy DD



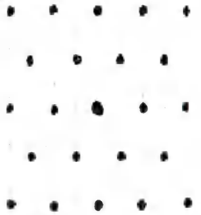
copy EE

figurations

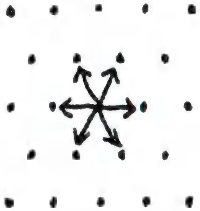
Warranty

red letters

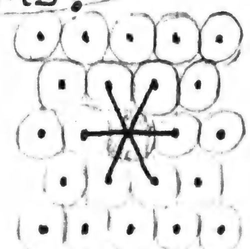
HOW TO PLAY WORMS?



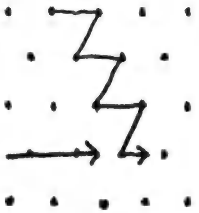
In the beginning, the screen is covered with dots. The worms start in the center.



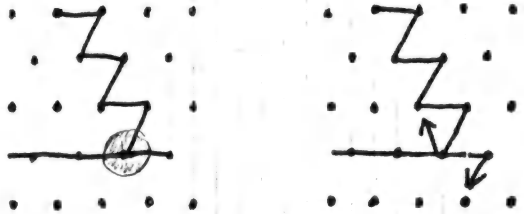
Worms move in one of 6 directions. They leave a trail behind. It is illegal to move along an existing trail.



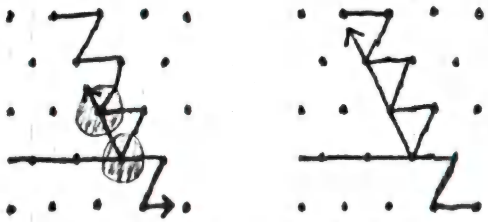
Each dot lies in the center of a "territory." Notice that a trail is half in one territory and half in another.



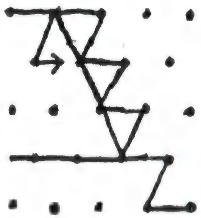
Once a worm recognizes a movement pattern, it will move automatically until it runs into something. [See HOW TO TRAIN YOUR WORM.]



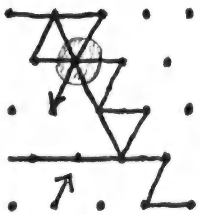
At this point, the black worm needs to be told what to do using the appropriate movement command. Check the bottom of the screen to determine which worm needs direction.



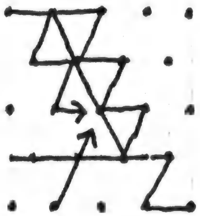
when a worm adds a new trail to a dot's territory, all other existing trails in the territory turn to that color.



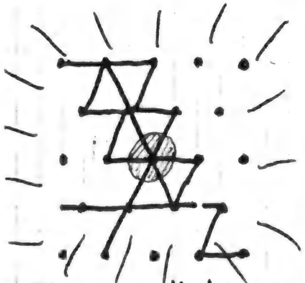
Several moves later, the red worm is coming back. It is changing the territory's color again.



ONE POINT is scored by the worm who adds the last (sixth) trail in the territory.
The territory flashes when the last trail is added



Red and black are headed for the same dot. Once at the dot, any move would be illegal.

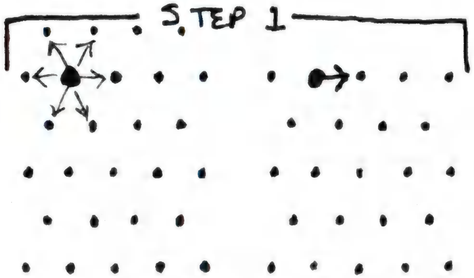


The worms die because they cannot move. Each worm's conquered territory flashes, the score is shown, and the game is over.

Maynard's Solitaire to go here

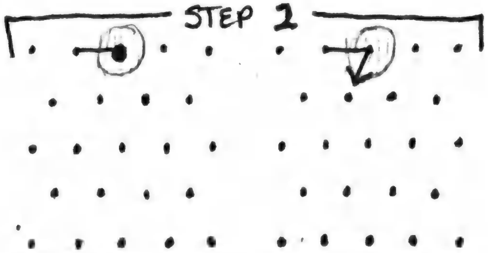
HOW TO TRAIN YOUR WORM

STEP 1



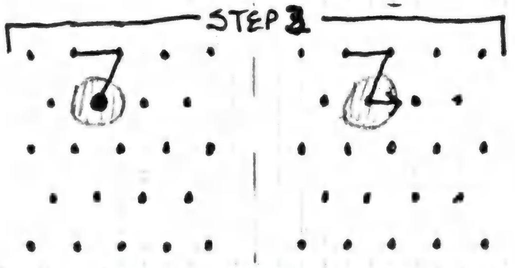
In the beginning, the worm starts in the center territory. (Pretend the red dot is in the center.) It can move in one of six directions, but it doesn't know which to choose. Let's train the worm to zigzag. Move it to the right.

STEP 2



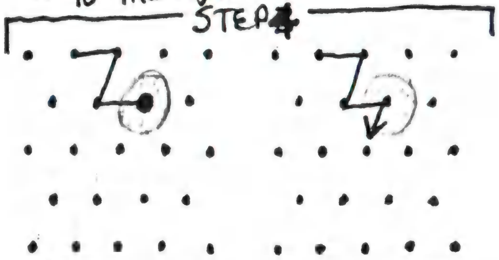
Now the worm is in a new territory with one trail going off to the left (—•). Move the worm to the lower left (↙). The worm will now remember that when it encounters —• it moves ↙.

STEP 3



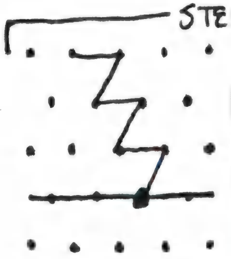
In this territory, there is a trail coming in from the upper right (↙). Move the worm to the right (→). Now it will remember that when it encounters ↙, it moves →.

STEP 4



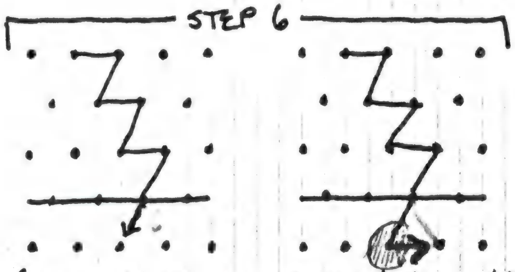
Since the worm remembers this configuration from step one, it remembers to move to the lower left. And so it continues...

STEP 5



... until it runs into something. The worm does not recognize the ⊥ configuration. The player must now tell the worm what to do.

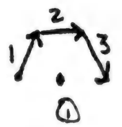
STEP 6

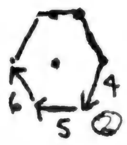


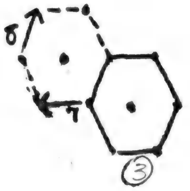
If you move the worm to the lower right, it encounters ↙ in the next territory. The worm recognizes this from Step 3, so it will move →.

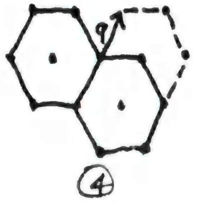
64 Configurations go here

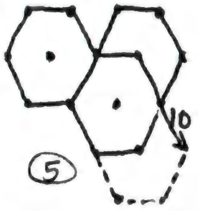
HOW TO CREATE MAYNARD'S SOLITAIRE

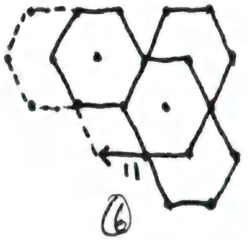


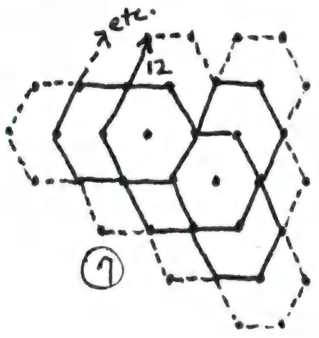








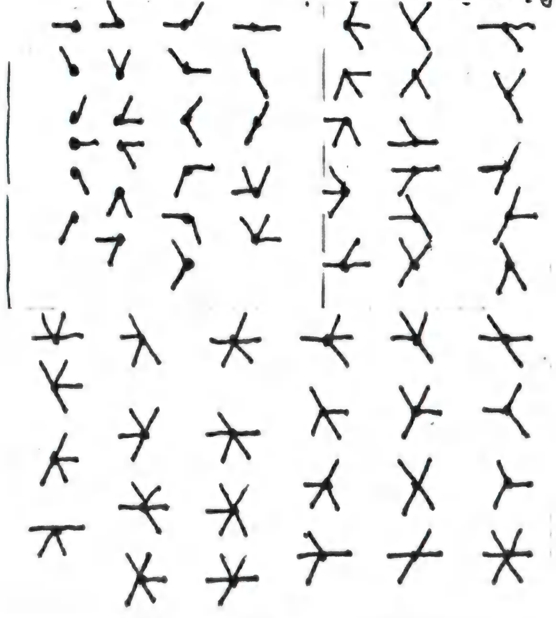




You must
use only
one NEW
worm (in
any color).
Follow the
12 moves shown in red.

THE 64 POSSIBLE TRAIL CONFIGURATIONS

• (Note that this configuration only appears during the very first move.)



Not all
these
configurations
will show up
in the same
game.

REFERENCE CARD
FOR WORMS?

To Begin:

OPTION key	Allows you to choose the type of worm (see below)
SELECT key	Moves you to the next worm. The SELECT/OPTION/START line is the same color as the selected worm.
START key	Begins game. In the middle of the game, START takes you back to the ~~OPTION~~ SELECT/~~SELECT~~ OPTION/START menu.

Use SELECT and OPTION only if you want to change worm types. Otherwise, just ~~press~~ use START

Worm Types:

NEW	The worm is untrained. You must train it as you play the game.
AUTO	Just like NEW, but the computer trains ~~it~~ the worm as the game goes along.
WILD	The computer trains the worm for all possible moves before the game begins
SAME	The same worm which was in the last game is used in this game. NEW and AUTO worms can be used for SAME.
----	The worm is asleep and doesn't play

SEE ~~How~~ TO SAVE WORMS

To Move Worms

Keyboard	• use space bar to position worm • ~~hit~~ press any key to leave trail
Paddle	• use paddle to position worm in 1 of 6 directions • press red fire button to leave trail

Keyboard movements and Paddle movements can be used in same game.

Worms
RC-2

OTHER ~~[illegible]~~

keys 1 to 9	sets speed of the worms 1 is slow 9 is fast Speed can be set any time during the game.
ESC key	halts game; press key again to resume game

See To SAVE WORMS for the following commands

W ~~key~~	writes worm ~~to disk~~ onto separate disk
R	reads worm from separate disk
U	updates already written worm
D	directory; lists saved worms

To SCOR~~ING~~E 1 point goes to the worm that ~~puts~~ lays the last (sixth) trail in a territory

To SAVE WORMS

Preparing

You must format a blank disk and write Atari DOS onto that disk. When saving worms and using saved worms, first load the game program into the Atari. Then, remove the game program disk from the drive and insert the blank, formatted disk (with the Atari DOS on it) into the drive.

Worms
RC-3

Writing Worms

- Use the SELECT key to indicate which worm you want to SAVE.
- Press W for WRITE. The computer will respond WRITE? <RETURN>
- Press RETURN key for "yes." If you make a mistake and don't want to write that worm, press any other key to return to the SELECT/OPTION/START mode.
- The computer will ~~respond~~ ask FILE NAME: Enter a worm name that is no more than 4 characters long. (You might use your initials and numbers). Then press RETURN.

The screen will go blank as the worm is saved onto disk.

Reading Worms

The instructions are very similar to Writing Worms, above. Make sure you have the disk with the saved worms in the disk drive.

- Use the SELECT key to indicate which ~~worm will saved~~ color ~~you~~ want your saved worm ~~will be~~. to play.
- Press R for READ. The computer will respond READ? <RETURN>
- Press RETURN and the computer will ask FILE NAME:
- Enter the worm's name, and press RETURN.

The screen will go blank as the worm is loaded.

Updating Worms

When you are playing with a saved worm and you are teaching it new moves, you may want the updated moves saved. Again, make sure you have the disk with the saved worms in the disk drive.

Worms
RC-4

- Press the SELECT key to choose which worm is being updated.
- Press ~~the~~ U for UPDATE, then press RETURN.

The screen will go blank while the new moves are being added to the disk

Directory

Have your disk with the saved worms in the disk drive.

- Press D for DIRECTORY, then press RETURN. The computer will respond DIRECTORY? ‹RETURN›
- Press ~~the~~ RETURN, and the names of your saved worms will be listed at the top of the screen.
- ~~Hit~~ Press OPTION or SELECT ~~or START~~ to put you back in the SELECT/OPTION/START menu. After you've made your choices and press START, the ~~g~~ directory will be replaced with the playing field.

Another WORMS? manual draft, this one replaces the last. This one is also reasonably to scale in terms of the actual finished size.

Susan

Susan

Nice job!

I have some notes on how you might "introduce" the game, but your style is excellent.

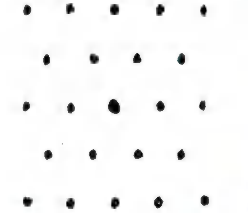
Thx

T

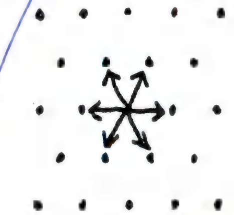
This seems somewhat esoteric to introduce early — I think it would be better to quickly outline the game objectives FIRST:

0. How to START the game
1. Capture the most dots
    1.1 Dot captured when 6th line drawn
2. Program your worm
3. Worms move until they "crash"
4. How to know when it's your turn to move
5. How to input a move

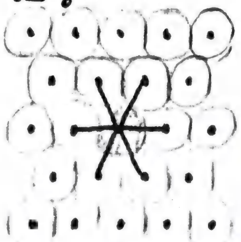
HOW TO PLAY WORMS?



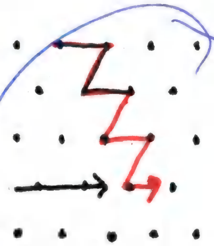
In the beginning, the screen is covered with dots. The worms start in the center.



Worms move in one of 6 directions. They leave a trail behind. It is illegal to move along an existing trail.

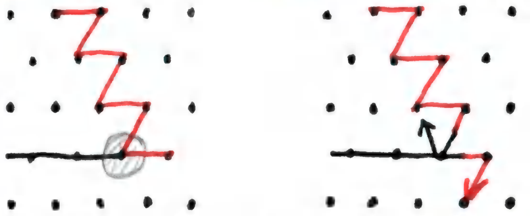


Each dot lies in the center of a "territory." Notice that a trail is half in one territory and half in another.



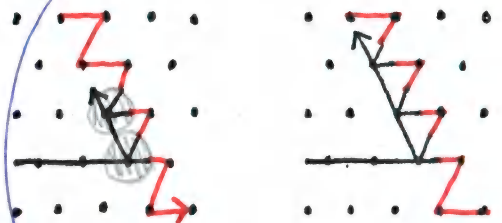
good

Once a worm recognizes a movement pattern, it will move automatically until it runs into something. [See HOW TO TRAIN YOUR WORM.]

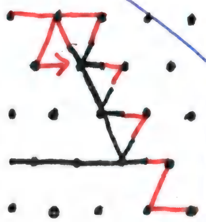


At this point, the black worm needs to be told what to do using the appropriate movement command. Check the bottom of the screen to determine which worm needs direction.

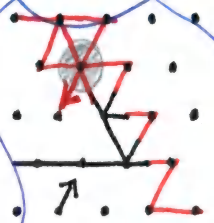
?



when a worm adds a new trail to a dot's territory, all other existing trails in the territory turn to that color.



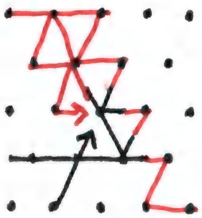
Now the red worm is coming back, changing the territory's color again.



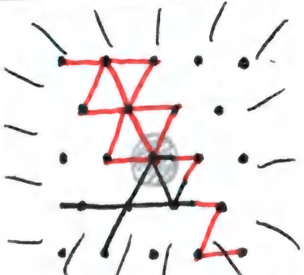
good

ONE POINT is scored by the worm who adds the last (sixth) trail in the territory.

Notice the black worm moving up the bottom.

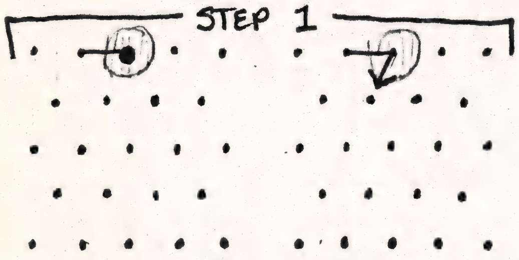


Red and black are headed for the same dot. Once at the dot, any move would be illegal.

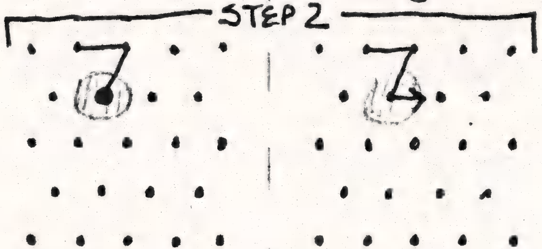


The worms die because they cannot move. When all the worms are dead Each worms conquered territory flashes, the score is shown, and the game is over.

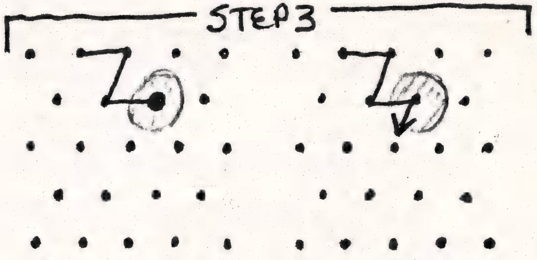
HOW TO TRAIN YOUR WORM, using the zigzag pattern as an example



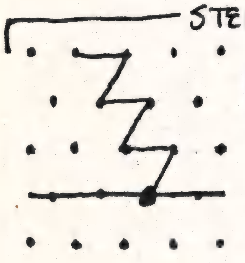
If there is only one trail off to the left, the player moves the worm to the lower left. The worm will remember this move; whenever it runs into a —• configuration, it will go ↙



If there is only one trail coming in from the upper right, the player mover moves the worm to the right. The worm will remember this, so that whenever it runs into a ⌐ configuration, it will go ↳



Since the worm remembers this configuration from step one, it remembers to move to the lower left. And so it continues . . .

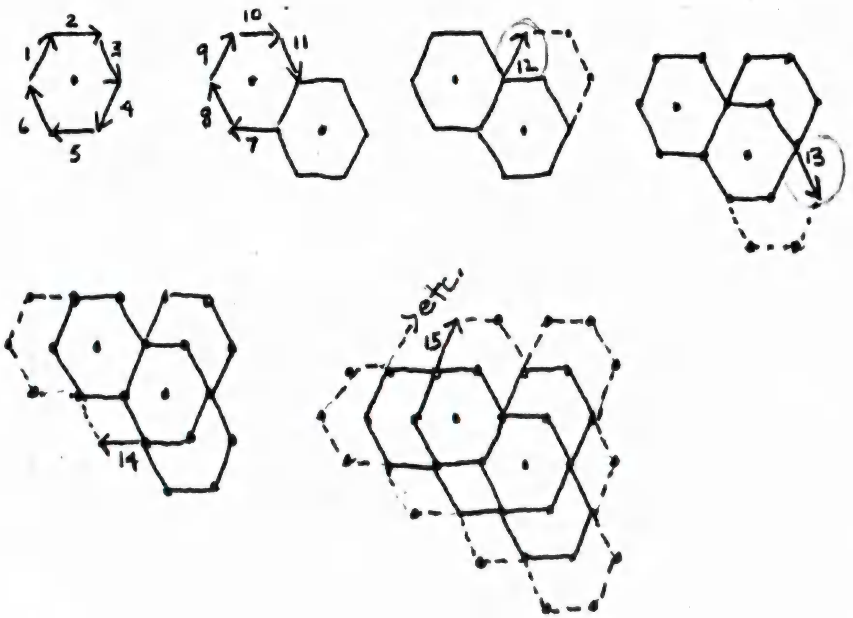


. . . until it runs into something. The worm does not recognize the ⊥ configuration. The player must now tell the worm what to do,

awkward. clearer to say something like "until the worm moves into another path or configuration that it does not recognize.

There are 64 possible configurations that a worm can encounter. And here they are:

NOTE: The worm only pays attention to the trail configuration; not the color of the this is great!



Follow the 15 steps above, ~~a~~ using one NEW worm, and you can create Maynard's solitaire, shown on the package

HOW TO CREATE MAYNARD'S SOLITAIRE

(Title page)

REFERENCE CARD FOR WORMS?

This is a nice summary page - well written

TO BEGIN:

OPTION key	Allows you to choose the type of worm (see below)
SELECT key	Moves you to the next worm.
START key	Begins game — at any time, even in the middle of another game

WORM TYPES:

NEW	The worm is untrained. You must train it as you play the game.
AUTO	Just like NEW, but the computer trains and plays the worm
WILD	The computer moves the worm randomly. The worm cannot be trained.
SAME	The ~~[illegible]~~ same worm is used in the next game. NEW and AUTO worms can be used for SAME.
----	The worm is asleep and doesn't play.

Should this be on a separate reference card/warranty card — or should it be a 4th panel on the instructions?

Separate card would be a hassle - how about the 4th panel?

TO MOVE WORMS:

Keyboard
- use space bar to position worm
- press RETURN key to leave trail

Paddle
- use paddle to position worm
- ~~use~~ press red fire button to leave trail

OTHER COMMANDS:

keys 1 through 9	sets the speed of the worms. 1 is slow 9 is fast can be set at any time during the game
keys P or ESC	halts game; hit the key again to resume game

3.23.83 Pat

This is a rough draft of the manual for WORMS?. To be added include:

- SAVE WORM instructions
- synopsis of mathematical principles (maybe)
- directions for Maynard's solitaire (maybe)

Feedback would be appreciated.

Susan

~~P.S. The manual is~~

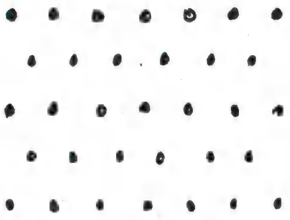
Susan —

I made a few comments. Given how complex the game is, I think you did a nice job of explaining it!

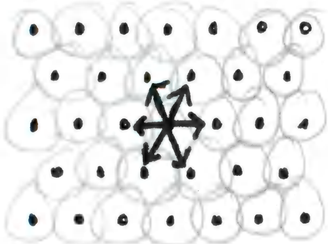
Pat

HOW TO PLAY

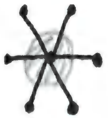
WORMS?



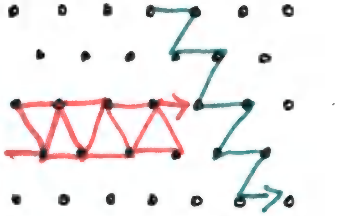
At the start of the game, the playing field is covered with dots.



- Each dot lies in the center of a "territory.
- Worms move from one dot to another in one of 6 directions, leaving a trails ~~in its wake~~.
- Worms cannot move over an existing trail.

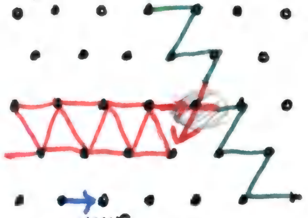


Note that a trail is half in one territory and half in another.



Once a worm has been trained, it will move automatically. It will stop only when it doesn't know what to do.
[Check the bottom of the screen to see which worm needs direction.]
When your worm has stopped, you need to give it its next move.

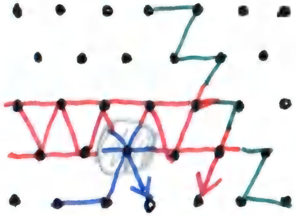
SEE HOW TO TRAIN YOUR WORM.



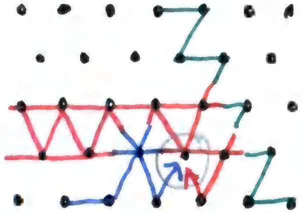
When your worm adds a trail to a dot, all other trails in the territory turn ~~the~~ ~~same color~~. to your color. (only 1/2)

existing?

(I never realized this!!)



When you add the last (sixth) trail to the territory, you earn 1 point.



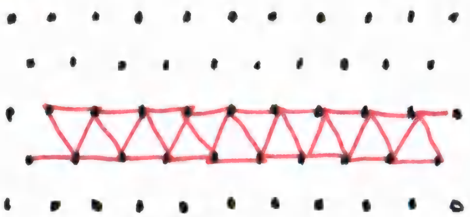
Worms "die" when they cannot make a legal move.

The game is over when all the worms are dead.

NOTE: If an even number of worms are playing, worms will die by running into each other. If an odd number of worms are playing, the odd worm dies at the same dot it originally started from.

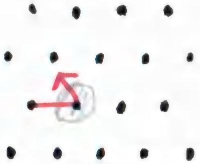
HOW TO TRAIN YOUR WORM

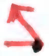
WORMS?



The worm recognizes the "trail" pattern at each dot. If the worm has encountered that particular pattern before, it will repeat the same move

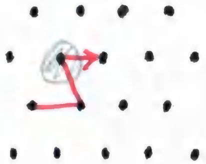
Using the triangular pattern, we will go through each step to determine when you, the player, move the worm and when the worm moves automatically.

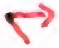




(A)

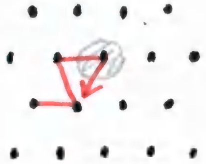
Player moves worm

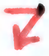




(B)

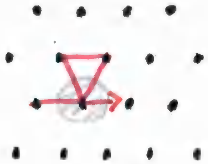
Player moves worm

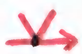




(C)

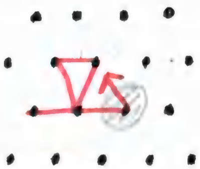
Player moves worm





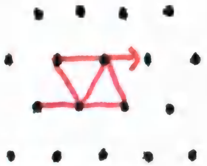
(D)

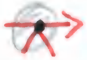
Player moves worm



[See pattern (A)]

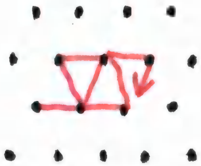
Worm moves automatically





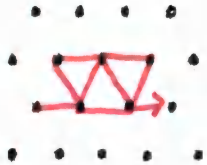
(E)

Player moves worm



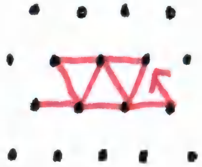
[See pattern (C)]

Worm moves automatically



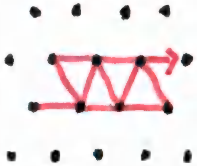
[See pattern (D)]

Worm moves automatically



[See pattern (A)]

Worm moves automatically

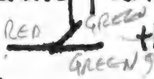


[See pattern (E)]

Worm moves automatically

The worm continues to move automatically, until ... ? it runs into something. it doesn't recognize

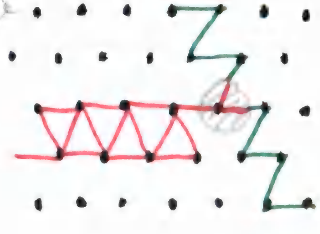
when this Now a new pattern is presented eg.



that the worm doesn't recognize.

The player must now move the worm.

NOTE: There are 64 possible patterns. Not all of them are encountered in any one game.



WORMS?

TO BEGIN:

OPTION key	Allows you to choose the Worm ~~Play~~ (see below)
SELECT key	Moves you to the next worm
START key	Begins game -- even if you are in the middle of another game

WORM PLAY:

Some confusion between play and train

Am I training the worm or is the computer?

NEW	You, the player, train the worm [See HOW TO TRAIN YOUR WORM] The worm is untrained. You must train it as you play the game.
AUTO	The computer trains the worm. Same as NEW, but the computer trains and plays the worm.
WILD	The worm moves randomly and does not recognize patterns
SAME	The worm moves the way it was trained in the previous game. — by you? or by computer? She may have to do some training during play.
----	The worm is asleep and doesn't play

TO MOVE WORMS:

pause = P key

Keyboard	• use space bar to position worm • hit ~~any other key~~ RETURN to register the move — leave the trail
~~Joystick~~	PRESS? • ~~move joystick to position worm~~ • ~~hit red fire button to register the move~~ — leave the trail
Paddle	• move paddle to position worm • hit red fire button to register the move
SPEED	• hit a number from 1 to 9 -- 1 is slow; 9 is fast when can I do this? any time?

Pat

Another WORMS? manual draft, this one replaces the last. This one is also reasonably to scale in terms of the actual finished size.

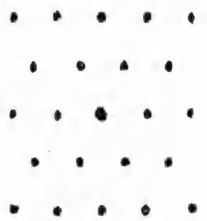
Susan

Susan —
I think this is more difficult to understand than the earlier version I saw.

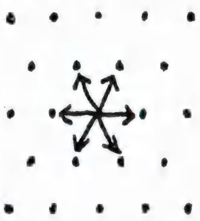
4 panels

Fine up to here — this assumes I know how to train the worm, but when I go to TRAIN it starts "IF — — —" Where is ground ϕ: i.e. this a dummy with 1 dot on the screen.

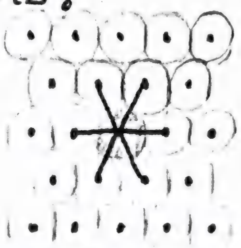
HOW TO PLAY WORMS?



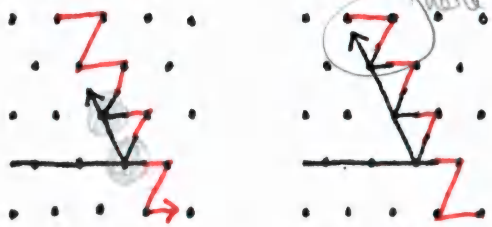
In the beginning, the screen is covered with dots. The worms start in the center.



Worms move in one of 6 directions. They leave a trail behind. It is illegal to move along an existing trail.

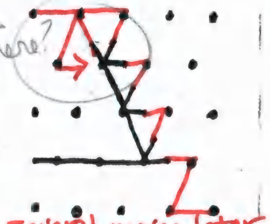


Each dot lies in the center of a "territory." Notice that a trail is half in one territory and half in another.

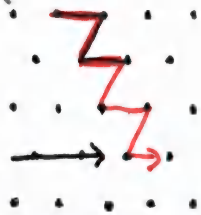


when a worm adds a new trail to a dot's territory, all other existing trails in the territory turn to that color.

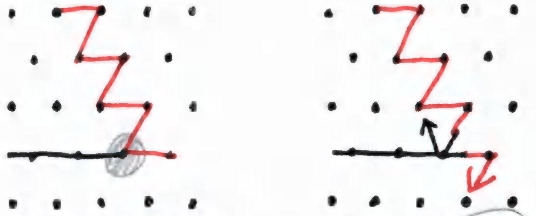
How'd I get from here to here?



Several moves later ~~Now~~ the red worm is coming back changing the territory's color again.

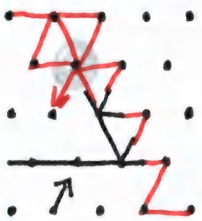


Once a worm recognizes a movement pattern, it will move automatically until it runs into something. [See HOW TO TRAIN YOUR WORM.]

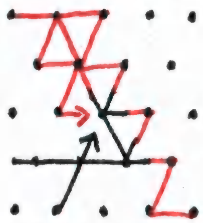


At this point, the black worm needs to be told what to do using the appropriate movement command. Check the bottom of the screen to determine which worm needs direction.

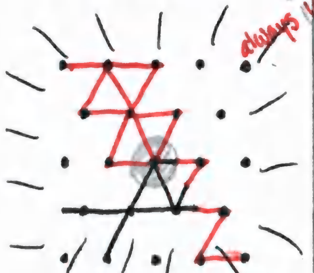
The ~~[illegible]~~ worm needing no direction ~~[illegible]~~ always Underlined! at the bottom of the screen.



ONE POINT is scored by the worm who adds the last (sixth) trail in the territory. The whole territory flashes, too. Notice the black worm moving up the bottom.



Red and black are headed for the same dot. Once at the dot, any move would be illegal.



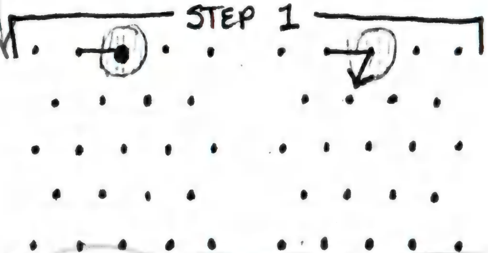
The worms die because they cannot move. Each worm's conquered territory flashes, the score is shown, and the game is over.

add Step at blank screen

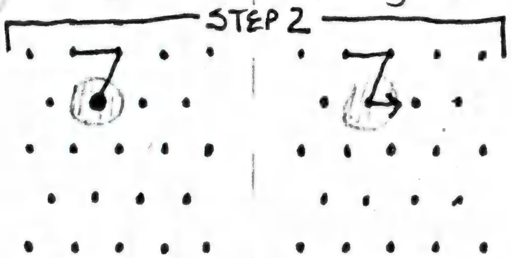
What is there isn't;

But we could also move it in some other direction. It's not clear this is an example. It doesn't show me the real basics; ie here's a dot & I can move my worm in any of 6 directions

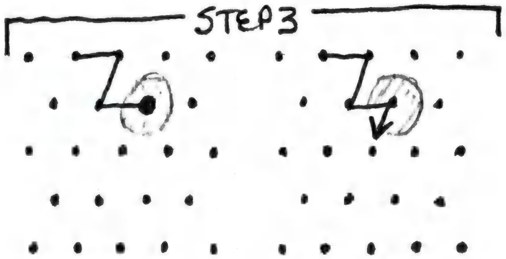
HOW TO TRAIN YOUR WORM, using the zigzag pattern as an example



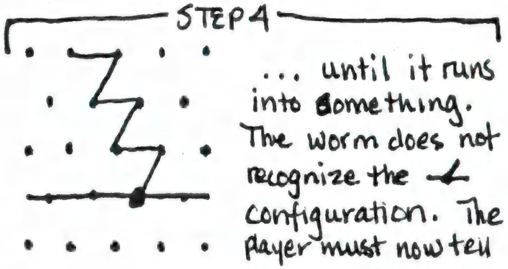
If there is only one trail off to the left, the player moves the worm to the lower left. The worm will remember this move; whenever it runs into a —• configuration, it will go ↙



If there is only one trail coming in from the upper right, the player mover moves the worm to the right. The worm will remember this, so that whenever it runs into a / configuration, it will go ↳



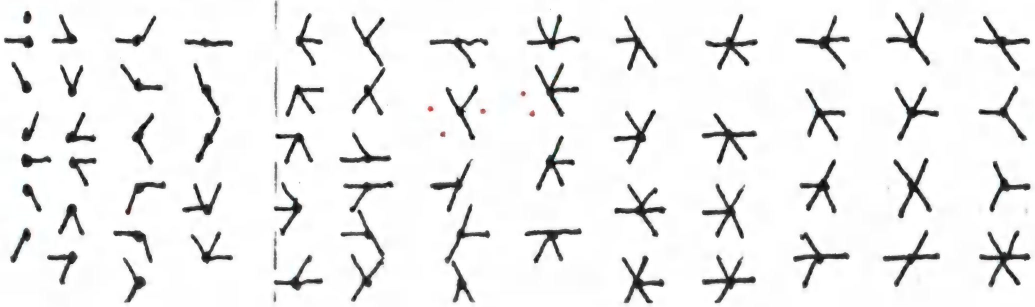
Since the worm remembers this configuration from step one, it remembers to move to the lower left. And so it continues...



... until it runs into something. The worm does not recognize the ⊥ configuration. The player must now tell the worm what to do,

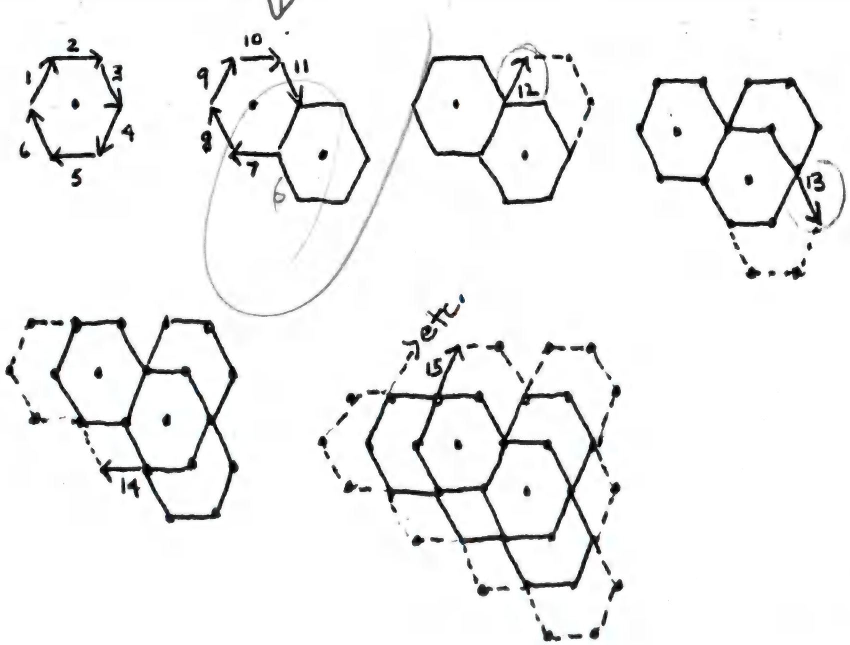
NOTE: The worm only pays attention to the trail configuration; not the color of the

There are 64 possible configurations that a worm can encounter. And here they are:



NOT ALL THESE CONFIGURATIONS WILL APPEAR IN THE SAME GAME.

rework #5



Follow the 15 steps above, using ~~a~~ one NEW worm, and you can create Maynard's solitaire, shown on the package

HOW TO CREATE MAYNARD'S SOLITAIRE

(Title page)

FINAL COPY

Susan -
FY files
- N

FRONT COVER:

WORMS?™ by David S. Maynard

(Logo)

Electronic Arts

~~INSIDE:~~ Back cover

Imagine a new kind of creature. Animated, but not alive. Synthetic, yet intelligent. A product of modern electrogenetic science.

Imagine that these creatures live in a world where reality is a two-dimensional grid. Without material bodies, they exist as light; pure energy. Without politics or religion, their lives are an unending adventure in pure mathematics. In music. In geometry.

Now, suppose they're worms.

How does their existence affect yours? Are you charmed, challenged, or careless in their presence?

What if you had to take responsibility for their lives? That you could reach into their minds to instruct them. To actually program their existence.

Then turn them loose in the world.

What then?

~~(Solitaire Screen Shot)~~
~~(Caption to be written)~~

~~(Red & Blue Screen Shot)~~
~~(Caption to be written)~~

~~(4 Worms Screen Shot)~~
~~(Caption to be written)~~

Inside (cont'd)

Game design and execution:	David S. Maynard
Package design:	Steinhilber, Deutsch & Gard
Linernotes:	Bill Morris
Photography:	Larry Keenan
Producer:	Susan W. Lee-Merrow

To
nterview

"~~Mr. Maynard~~ I would like to thank the following people: Steve Hayes for programming AUTO worms and the introduction music. Anson Averell, Jan Cornish, Phil Rozario, and Glenn Sherwood (The Sandbox) for creative consulting during development. This sentence would like to acknowledge Douglas Hofstadter for its existence. Finally ~~he~~ I, would like to dedicate this game to Martin Gardner who taught ~~him~~ me, and millions of others, that mathematics can be fun."

WORMS? is a ~~registered~~ trademark of Electronic Arts. Package design © Copyright 1983 Electronic Arts. Software © Copyright 1983 David S. Maynard. Simultaneously published in Canada and the U.S.A. Screen photos represent Atari 800 version only. Other versions may vary.

BACK COVER

Subhead: A way to think about thinking.

This is a unique kind of computer game, one for thinking people. There are no alien ships to blow up, no dots to eat, no animated apes to outwit. Worms? is more accurately an experiment in artificial intelligence, and you can run it any way you want. You can compete for the most territory, ~~points~~, or program to draw patterns. Or play music. You get the idea.

(4 Wilds
Screen Shot) (Solitaire
Screen shot)

(Logo)

Who we are. We're an association of electronic artists who share a common goal. We want to fulfill the potential of personal computing. That's a tall order. But with enough imagination and enthusiasm we think there's a good chance for success. Our products, like this game, are evidence of our intent. If you'd like to get involved, please write us at: Electronic Arts, 2755 Campus Drive, San Mateo, CA 94403.

Electronic Arts provides a limited ninety day warranty on the recording media. See limited warranty statement enclosed. The warranty does not apply to the software programs themselves, which are provided AS IS.

Oc
BG
DSM

David S. Maynard
The intelligence behind WORMS?

When David Maynard wrote his first program back in 1965, computer games were played in climate-controlled warehouses with decks of punch cards. The computer he used required enough power to turn the USS Nimitz into a headwind.

"When I entered UC Berkeley, I was a physics major, because there was no computer science department. In 1972 I worked on the first office automation project at Stanford Research Institute and wrote programs for the first mouse-based user interface."

Why WORMS?

"In late 1979 I was leafing through an issue of Scientific American when I ran across the Mathematical Games column written by Martin Gardner. In it he recounted the work of scientists at Cambridge and MIT who were investigating the patterns created by an idealized mathematical worm. Gardner was suggesting these worms as a novel way of teaching geometry. But it gave me other ideas."

The worms idea crawled around in Maynard's head for three years before he sat down at an Atari 800 and wrote his first computer game. Fifteen years as a computer scientist gave him the background to perfect the first game that doesn't talk down to either the computer or the players.

Will David be writing more games in the future?

"Yeah, I've got a few ideas in the back of my head."

Inside

May be rephrased

~~WORMS?~~

~~By David S. Maynard~~

OBJECT: The worms are trained to capture territories by laying trails from dot to dot. Each dot lies in the center of a territory.

SCORING: 1 point goes to the worm that lays the last (sixth) trail in a territory. Scores are shown on the bottom of the screen.

TYPES OF WORMS: There are five worms built into the program. You may also play with worms you have saved from previous games (see To Save Worms). The five worms are as follows:

NEW	The worm is untrained. You must train it as you play the game.
AUTO	This worm is just like NEW, but the computer trains the worm to make smart moves as it plays the game.
WILD	The computer trains the worm randomly for all possible moves before the game begins.
SAME	The worm which played this color in the last game is used again in this game.
-----	The worm is asleep and doesn't play.

STARTING: Choose the type of worm you want for each color that will be playing. Then start the game and begin training your worm! (See the Reference Card for the appropriate commands.)

~~© 1983 Electronic Arts, San Mateo, CA 94403~~

HOW TO PLAY WORMS?

(A) The screen is covered with dots. Each dot lies in the center of a territory.

(B) Worms start in the center and can move in one of 6 directions. See the Reference Card for movement commands.

(C) When worms move, they leave a trail behind. Notice a trail is half in one territory and half in another.

(D) Move your worm in the direction you want. NOTE: The worm cannot move along an existing trail. Once a worm recognizes a movement pattern, it will move automatically, until it runs into something (see How Worms Understand Their Training).

(E) The worm that needs a movement command is flashing, and its name is underlined at the bottom of the screen.

(F) When a worm adds a new trail to a dot's territory, all other existing trails in the territory turn to that color.

(G) Several moves later, the red worm is coming back. It is changing the territory's color again.

(H) The red worm has scored a point by laying the last (sixth) trail in the territory. The territory flashes when the point is scored.

HOW TO PLAY WORMS?

p.2

(I) Red and black are headed for the same dot. Once at the dot, any move would be illegal.

(J) The worms die because they cannot move. When all the worms are dead, each conquered territory flashes, the score is shown, and the game is over.

HOW ~~WORMS UNDERSTAND~~ THEIR TRAINING

(AA) ~~In the beginning~~, The worm starts in the center territory. (Pretend the red dot is in the center.) It can move in one of six directions, but it doesn't know which to choose. Let's train the worm to zigzag. Move it to the right.

(BB) Now the worm is in a new territory with one trail going off to the left (→). Move the worm to the lower left (↙). The worm will now remember that when it encounters → it moves ↙ .

(CC) In this territory, there is a trail coming in from the upper right (✓). Move the worm to the right (∠→). Now it will remember that when it encounters ✓ , it moves ∠→ .

(DD) Since the worm remembers this configuration from step one, it remembers to move to the lower left. And so it continues.....

(EE) ....until the worm moves into another path or configuration ~~it runs into something. The worm does not recognize the~~ (∠) it does not recognize, ~~configuration~~. The player must now tell the worm what to do.

(FF) If you move the worm to the lower right, it encounters ✓ in the next territory. The worm recognizes this from Step 3, so it will move ∠ .

BG
DSM

David S. Maynard

The intelligence behind WORMS?

When David Maynard wrote his first program back in 1965, computer games were played in climate-controlled warehouses with decks of punch cards. The computer he used required enough power to turn the USS Nimitz into a headwind.

" I entered UC Berkeley ~~I was~~ as a physics major, because there was no computer science department. But In 1972 I ~~worked~~ got to work on the first office automation project at Stanford Research Institute and wrote programs for the first mouse-based user interface, as part of the Augmentation Research Center. What were we augmenting? Human intelligence, of course!"

Why WORMS?

"In late 1979 I was leafing through a back issue of Scientific American when I ran across the Mathematical Games column written by Martin Gardner. He recounted the work of scientists at Cambridge and MIT who were investigating the patterns created by an idealized mathematical worm. Gardner was suggesting these worms as a novel way of teaching geometry. But ~~it gave me~~ to me it suggested other ideas."

The worms idea crawled around in Maynard's head for three years before he sat down at an Atari 800 and wrote his first ~~computer~~ game. Fifteen years as a computer scientist (and a personality prone to endless jokes, juggling and a high degree of playfulness) gave him the background ~~to~~ perfect WORMS?. ~~game that doesn't talk down to either the computer or the players."~~

"I think of the game as Zen and the Art of Worm-Training."

Will David be writing more games in the future?

"Sure! I've had ~~really~~ great jobs at places like SRI and Xerox, where I ~~got to~~ worked on pretty sophisticated projects. Now I ~~get to~~ can apply that knowledge and expertise to products that are really fun."

~~"Yeah, I've got a few ideas in the back of my head."~~

Maynard also gets to provide that fun to his family. His eight-year-old son and friends spend countless hours training worms and doing battle. Imagine what they will play next.

"I've got a few ideas in the back of my head."

BACK

~~INSIDE FRONT~~ COVER

Prelim

Subhead: A unique experience ~~A way to think about thinking~~.

~~This is a unique kind of computer game, one for thinking people.~~ There are no alien ships to blow up, no dots to eat, no animated apes to outwit. Worms? is more accurately an experiment in color, sound, and motion, ~~artificial intelligence~~ and you can run it any way you want. You can ~~compete for the most territory/points, or~~ ~~program to~~ draw patterns. Or play music. Or create sound-and-light shows. You get the idea.

But if you like challenges. If you dream of world domination. Then play the game.

Subhead: A game of territorial conquest.

Worms can be trained . . . to conquer. As they move, ~~among~~ they capture and recapture territory, until the last trail is laid. Can you train your worm to beat other players? To beat the computer itself?

Can you meet the challenge?

~~INSIDE~~

~~FRONT COVER~~:

OK
BG

WORMS? by David S. Maynard

INSIDE COVER

~~BACK COVER~~:

Imagine a new kind of creature. Animated, but not alive. Synthetic, yet intelligent. A product of modern electrogenetic science.

Imagine that these creatures live in a world where reality is a two-dimensional grid. Without material bodies, they exist as light; pure energy. Without politics or religion, their lives are an unending adventure in pure mathematics. In music. In geometry.

Now, suppose they're worms.

How does their existence affect yours? Are you charmed, challenged, or careless in their presence?

What if you had to take responsibility for their lives? That you could reach into their minds to instruct them. To actually program their existence.

Then turn them loose in the world. To conquer. To chase each other. To capture and recapture territory--trailing color and music as they move.

What then?

Other Worms? copy

Cover

WORMS? by David S. Maynard

Credits

Game design + execution : David S. Maynard
Package design : Barry Deutsch
Linernotes : Bill Morris
Photography : TBD
Producer : Susan Lee-Merrow

~~Mr. Maynard should~~
~~The [illegible]~~

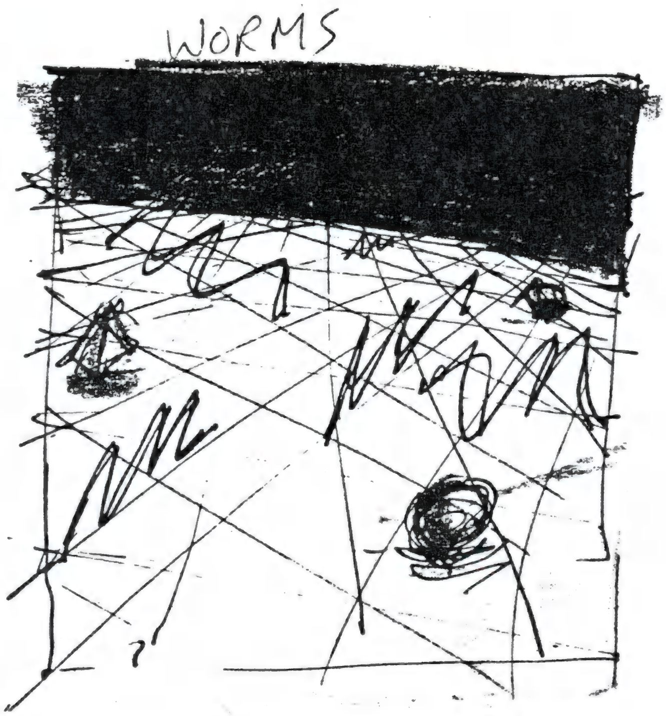
stet

~~I would~~ like to thank the following people: ~~for the contributions they have made to the development of this game.~~ Steve Hayes for programming the AUTO worms, ~~and the introduction music~~, Anson Averell, Jan Cornish, Phil Rozario, and Glenn Sherwood (The Sandbox) for creative consulting during development. This sentence would like to acknowledge Douglas Hofstadter for its existence. Finally I would like to dedicate this game to Martin Gardner who taught me, and millions of others, that mathematics can be fun.

there will be no introduction music

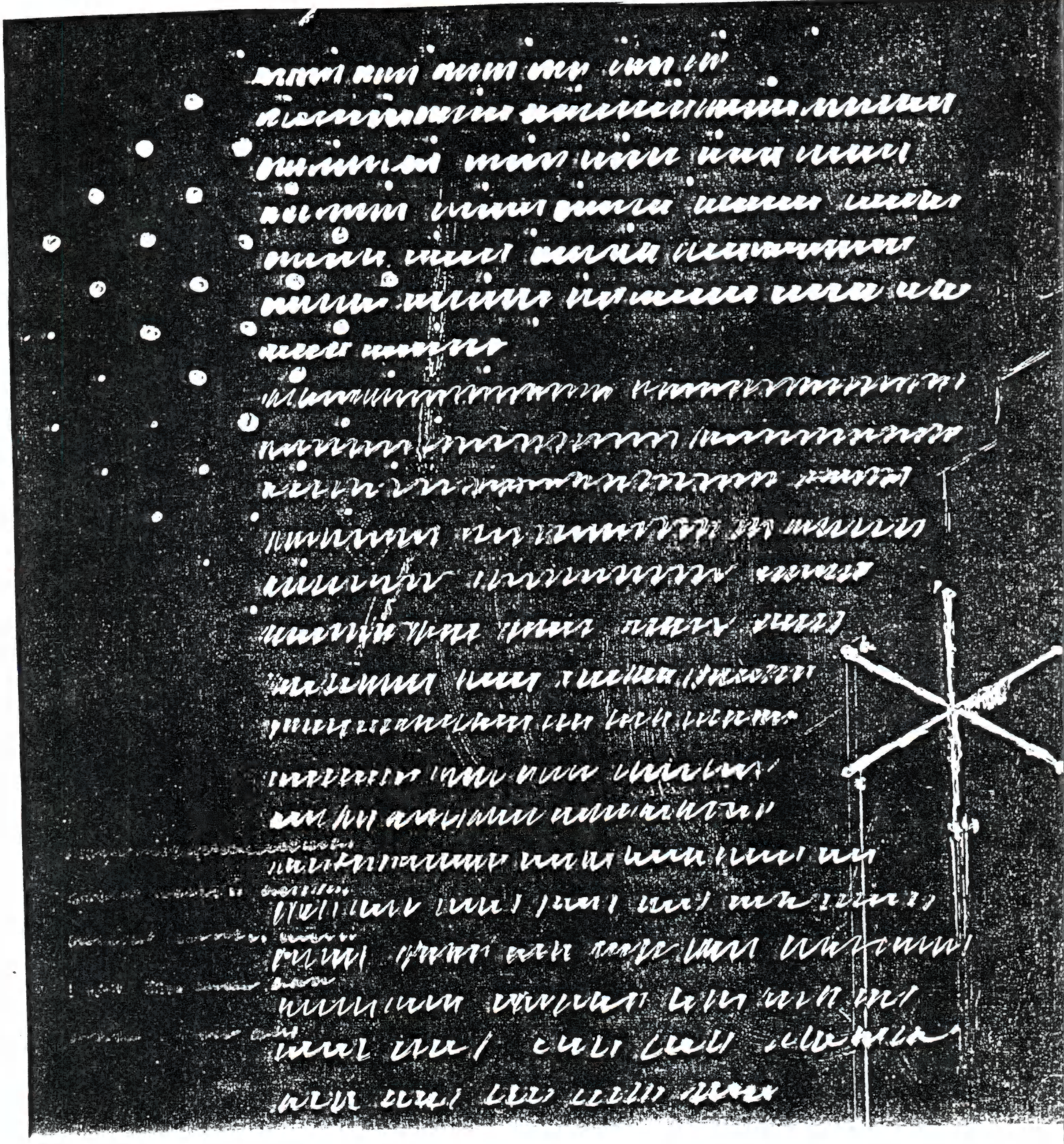
Legal / Copyright
EA copy
Screen disclaimer (Atari)
Warranty

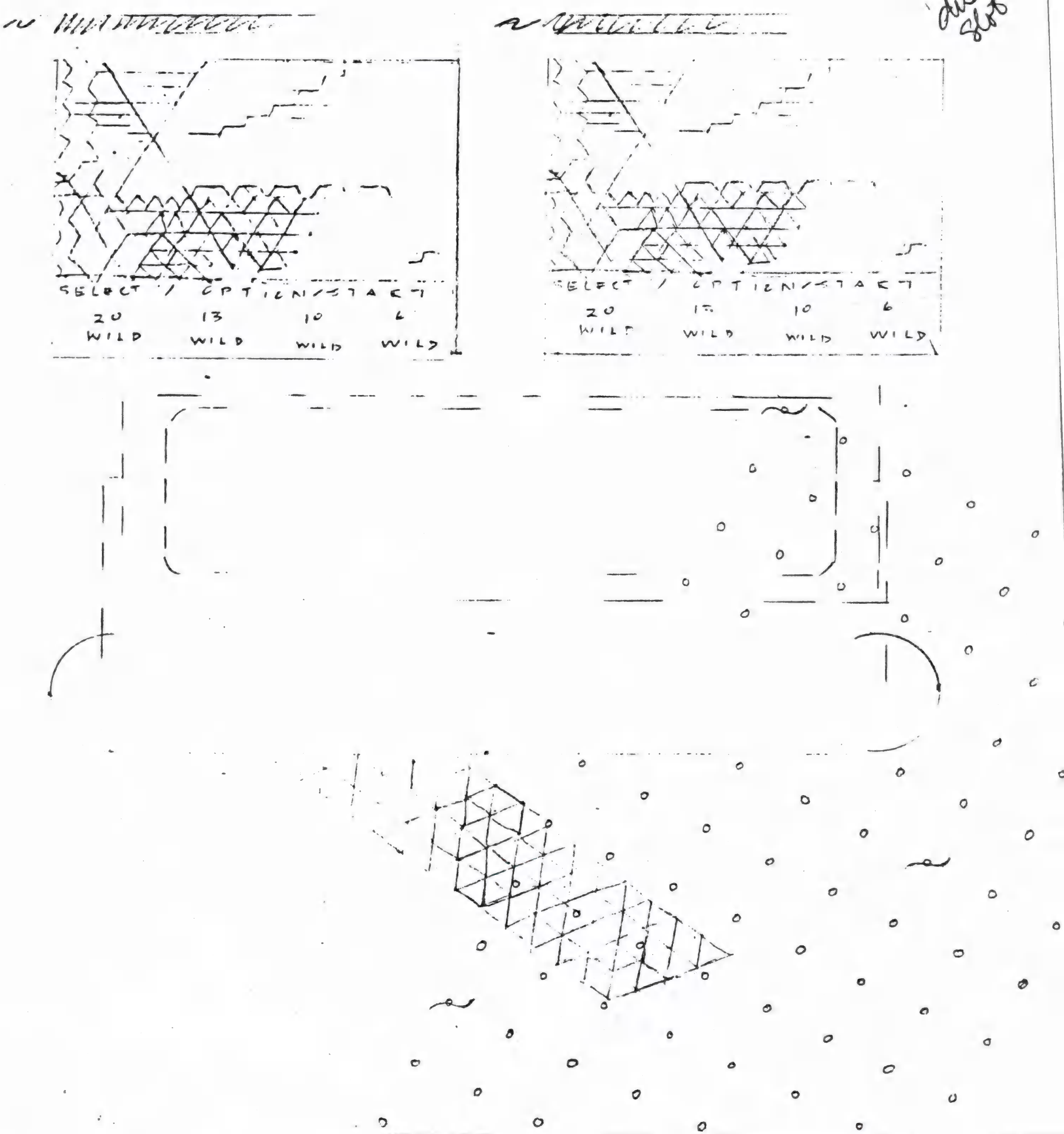
OK
BG



front

inside copy





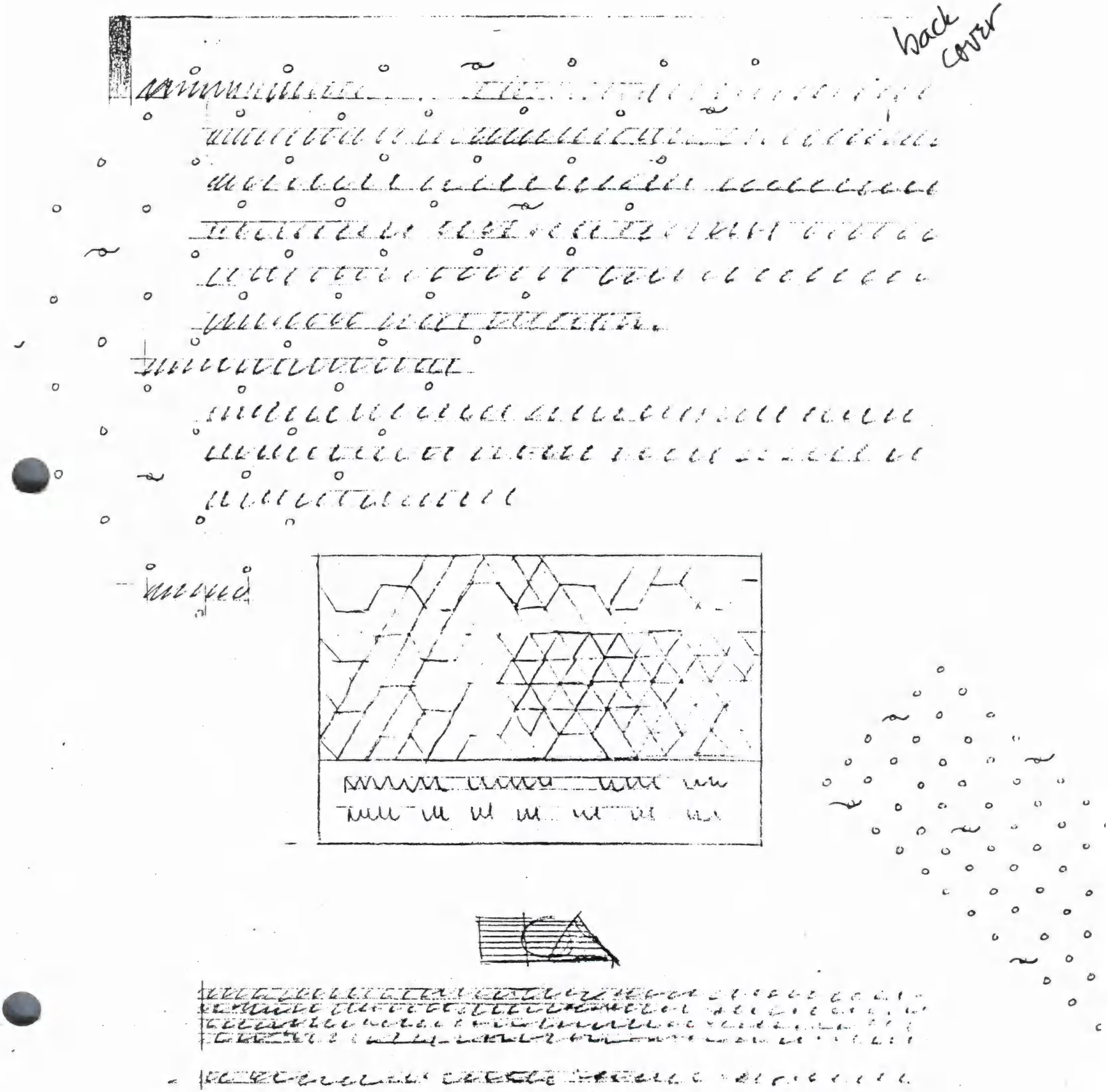
SELECT / OPTION/START
20 WILD
13 WILD
10 WILD
6 WILD
SELECT / OPTION/START
20 WILD
15 WILD
10 WILD
6 WILD

disk slot

picture

ok

back cover



INSIDE FRONT COVER

Subhead: ~~Back to earth~~ ~~But seriously~~, Use your head.

~~Nurdworms~~ Gridworms is a unique truly original kind of computer game, one for thinking people. There are no alien ships to blow up, no dots to eat, no animated apes to outwit. ~~Nurd~~ Gridworms is ~~the first artificially intelligent computer game, and is~~ as challenging to figure out as your own mind.

Subhead: How to play.

After booting the disk and pushing START, four worm directories will appear at the bottom of the screen. Each has five species of worms you control with the SELECT and OPTION keys.

The ~~NURD~~ GRID worm responds to your paddle programming. When the worm stops, steer it with the paddle, then press and release the button. And remember that move, because the worm will. G~~ive it the same instruction twice, and it will repeat that move~~ Whenever ~~it~~ the worm encounters that same situation. (configuration of existing spokes), it will repeat what it did last time in that situation.

AUTO is a smart worm controlled by the computer. When you play against AUTO, you're matching wits with the artificial intelligence of the computer.

WILD is a deranged worm, a mutant with unpredictable, sometimes unbeatable moves.

SAME makes the worm in question repeat the program of the preceeding games. When one worm is SAME, you're free to choose any species for the other three.

this option is being deleted

~~A worm in HAND does exactly what it's told and doesn't remember a thing.~~

* when ~~it's your turn, your~~ you need to tell your worm what to do, your color in text window flashes & is underlined

* set speed using # keys: 1-9 (no music with 9, it's too fast

* explain the "spokes of the wheel" mechanism

- whenever a new spoke is added, that color gets all the other spokes
- when you add the 6th spoke, you get a point
- the worm ~~discerns the pattern~~ decides the next move ~~by~~ by how many and where the existing spokes are

* describe illegal moves

(Game instructions continued)

head: The object(?) of the game.

~~Nurd~~Gridworms is an experiment in artificial intelligence, and you can run it any way you want. You can compete for the most territory/points, or program to draw patterns. Or play music. You get the idea.

Front Cover:

WORMS? by David S. Maynard

Back Cover:

Imagine a new kind of creature. Animated, but not alive. Synthetic, yet intelligent. A product of modern electrogenetic science.

Imagine that these creatures live in a world where reality is defined by a two-dimensional grid. Without material bodies, they exist as pure light energy. Without politics or religion, their lives are an adventure in pure mathematics. In music. In geometry.

Picture them as worms.

How does their existence affect your? Are you charmed, challenged, or careless in their presence.

Suppose these worms represented the first artificially intelligent game.

What then?

send in trained worms?
— contest?

Save game
NO unless
contest

Worms/Bing 3.24

- pkg - tomorrow
  pkg copy -

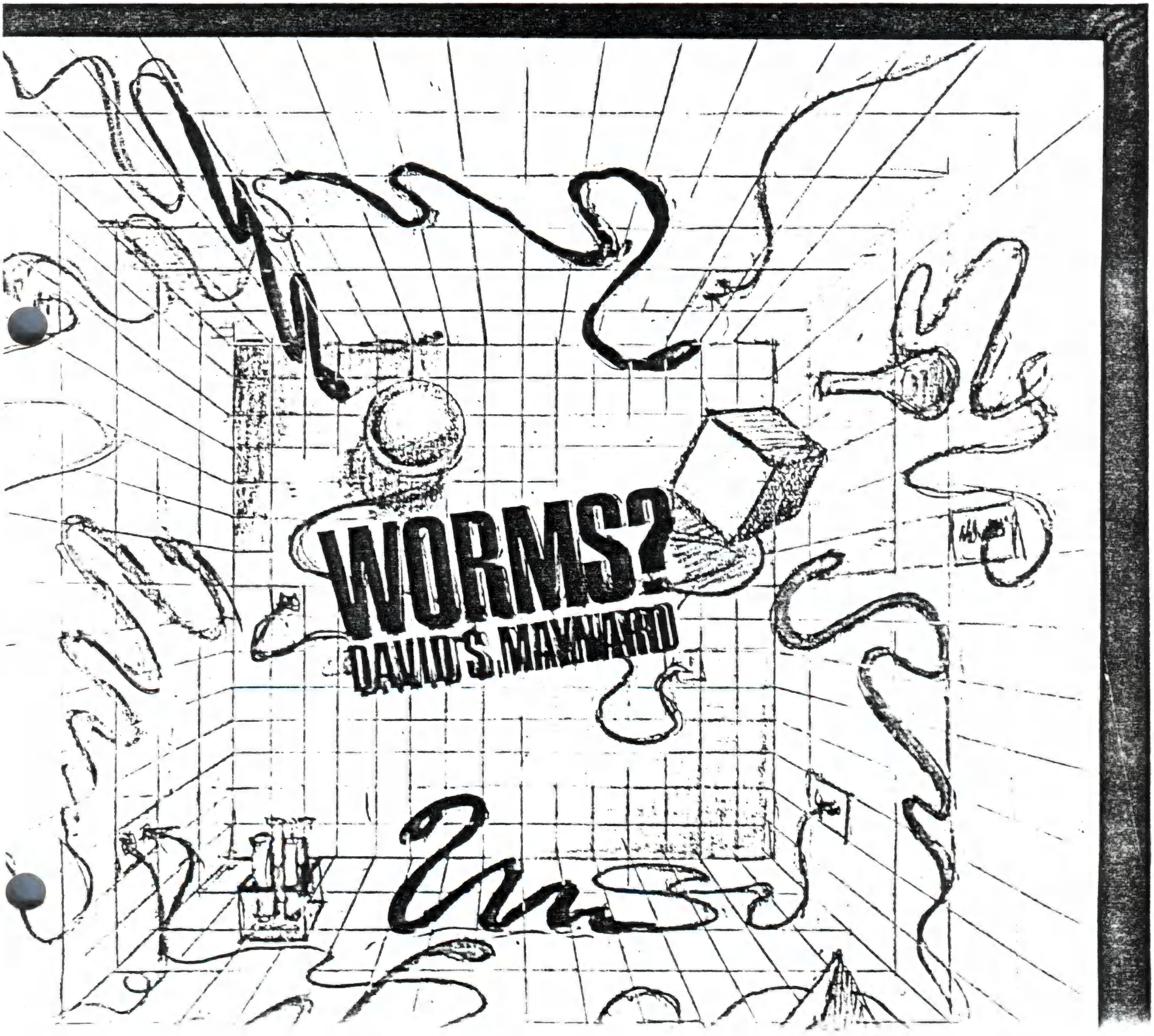
Maynard's Solitaire - yes
no math ~~stut~~ stuff
page of 64 permutations

2 color ink
ask Nancy about screens/shades

3-4 pg. gatefold

5¼ sq.

p14 – [illegible] cube, tetra [illegible]



FRONT COVER:

WORMS? by David S. Maynard

BACK COVER:

Imagine a new kind of creature. Animated, but not alive. Synthetic, yet intelligent. A product of modern electrogenetic science.

Imagine that these creatures live in a world where reality is a two-dimensional grid. Without material bodies, they exist as light; pure energy. Without politics or religion, their lives are an unending adventure in pure mathematics. In music. In geometry.

Now, suppose they're worms.

How does their existence affect yours? Are you charmed, challenged, or careless in their presence?

What if you had to take responsibility for their lives? That you could reach into their minds to instruct them. To actually program their existence.

Then turn them loose in the world.

What then?

INSIDE FRONT COVER

Subhead: A way to think about thinking.

This is a unique kind of computer game, one for thinking people. There are no alien ships to blow up, no dots to eat, no animated apes to outwit. Worms? is more accurately an experiment in artificial intelligence, and you can run it any way you want. You can compete for the most territory/points, or program to draw patterns, Or play music. You get the idea.

outside

make of wider interest

Subhead: How to play.

~~After booting the disk and pushing START, four worm directories will appear at the bottom of the screen. Each has five species of worms you control with the SELECT and OPTION keys.~~

~~The NEW worm responds to your paddle programming. When the worm stops, steer it with the paddle, then press and release the button. And remember that move, because the worm will. Give it the same instruction twice, and it will repeat that move whenever it encounters that situation.~~

~~AUTO is a smart worm that is programmed by the computer.~~

~~WILD is a deranged worm, a mutant with unpredictable, sometimes unbeatable moves.~~

~~SAME makes the worm in question repeat the program of the preceeding game.~~

~~A worm in HAND does exactly what it's told and doesn't remember a thing.~~

(Further playing suggestions will accompany EA-supplied screen photos.)

David S. Maynard
The intelligence behind WORMS?

When David Maynard wrote his first program back in 1965, computer games were played in climate-controlled warehouses with decks of punch cards. The computer he used required enough power to turn the USS Nimitz into a headwind.

"When I entered UC Berekely I was a physics major, because there was no computer science department." His first job was working for Lockheed in the flight test program for their L-1011 jumbo jet.

leave out

add SRI - Doug Englebart

augmentation research center augmenting human intellect

the father of [illegible] auto [illegible]

Why WORMS?

"In late 1979 I was leafing through an 1972 issue of Scientific American, when I ran across the Mathematical Games column written by Martin Gardner. In it he recounted the work of scientists at Cambridge and MIT who were investigating the patterns created by an idealized mathematical worm. Gardner was suggesting these worms as a novel way of teaching geometry. But it gave me other ideas."

The worms idea crawled around in Maynard's head for three years before he sat down at an Atari 800 and wrote his first computer game. Fifteen years as a computer scientist gave him the background to perfect the first game that doesn't talk down to either the computer or the players."

Will David be writing more games in the future?

"Yeah, I've got a few ideas in the back of my head."

BRK
*

I would like to thank the following people for the contributions they have made to the development of this game. Steve Hayes for programming the AUTO worms, and the introduction music. Anson Averell, Jan Cornish, Phil Rozario, and Glenn Sherwood (The Sandbox) for creative consulting during development. This sentence would like to acknowledge Douglas Hofstadter for its existence. Finally I would like to dedicate this game to Martin Gardner who taught me, and millions of others, that mathematics can be fun.

Acknowlidgaments
for WORMS?

ELECTRONIC ARTS
500 AIRPORT BOULEVARD
SUITE 440
BURLINGAME, CALIFORNIA
94010
(415) 340-0454

phoned 4.7.83
said it was OK
to use his name
Sm

March 22, 1983

Martin Gardner, President
Woods End, Inc.
103 Woods End Drive
Hendersonville, North Carolina 28739

Dear Martin Gardner:

In January, Pat Marriott wrote you describing the computer game WORMS? that we are planning to publish. At that time, she asked if we could use your name when we publish the game.

The specific quote would appear on the package, as part of an interview with the author of the game, David S. Maynard. The exact wording would be as follows:

> "In late 1979 I was leafing through a 1972 issue of Scientific American, when I ran across the Mathematical Games column written by Martin Gardner. In it he recounted the work of scientists at Cambridge and MIT, who were investigating the patterns created by an idealized mathematical worm. Gardner was suggesting these worms as a novel way of teaching geometry. But it gave me other ideas."

If this quote is acceptable to you, could you please drop me a note to that effect? (You could just make a copy of this letter, with an indication that it is OK, then mail it back.)

The game WORMS? will run on an Atari 400, 800, or 1200, when it is released at the end of May. In September we expect it to be available on the Commodore 64. If you have any of these machines or if you would simply like a copy of the product, I would be most pleased to send you one.

By the way, Conway's Game of Life has been done numerous times on the computer. In fact, these programs are in the public domain. Atari is selling a version of it on their Video Easel cartridge.

I look forward to hearing from you. If you have any questions about this, please don't hesitate to call me.

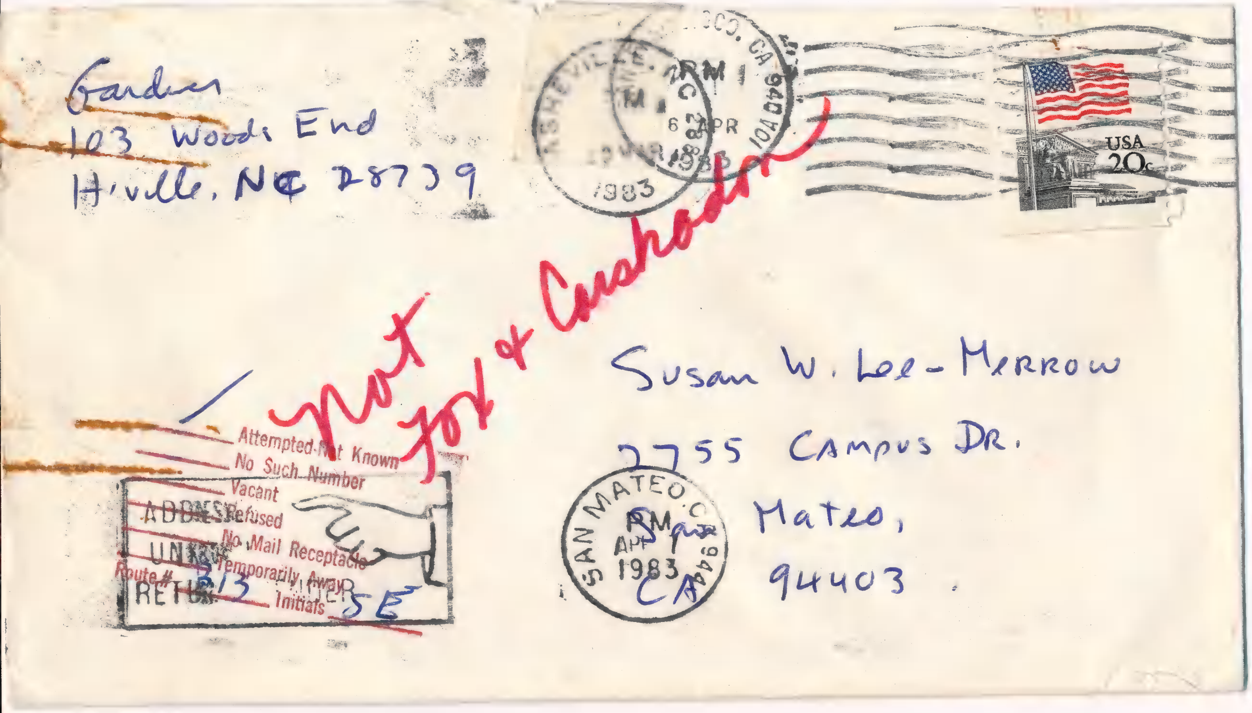
Sincerely,

Susan W. Lee-Merrow

Susan W. Lee-Merrow
Producer

P.S. We have moved. Our new address is:

NEW ADDRESS: 2755 CAMPUS DRIVE
SAN MATEO, CA 94403
(415) 571-7171



Gardner
103 Woods End
H'ville, NC 28739

ASHEVILLE, NC
1983

...CO. CA 940
PM 1
APR 6
1983

USA 20c

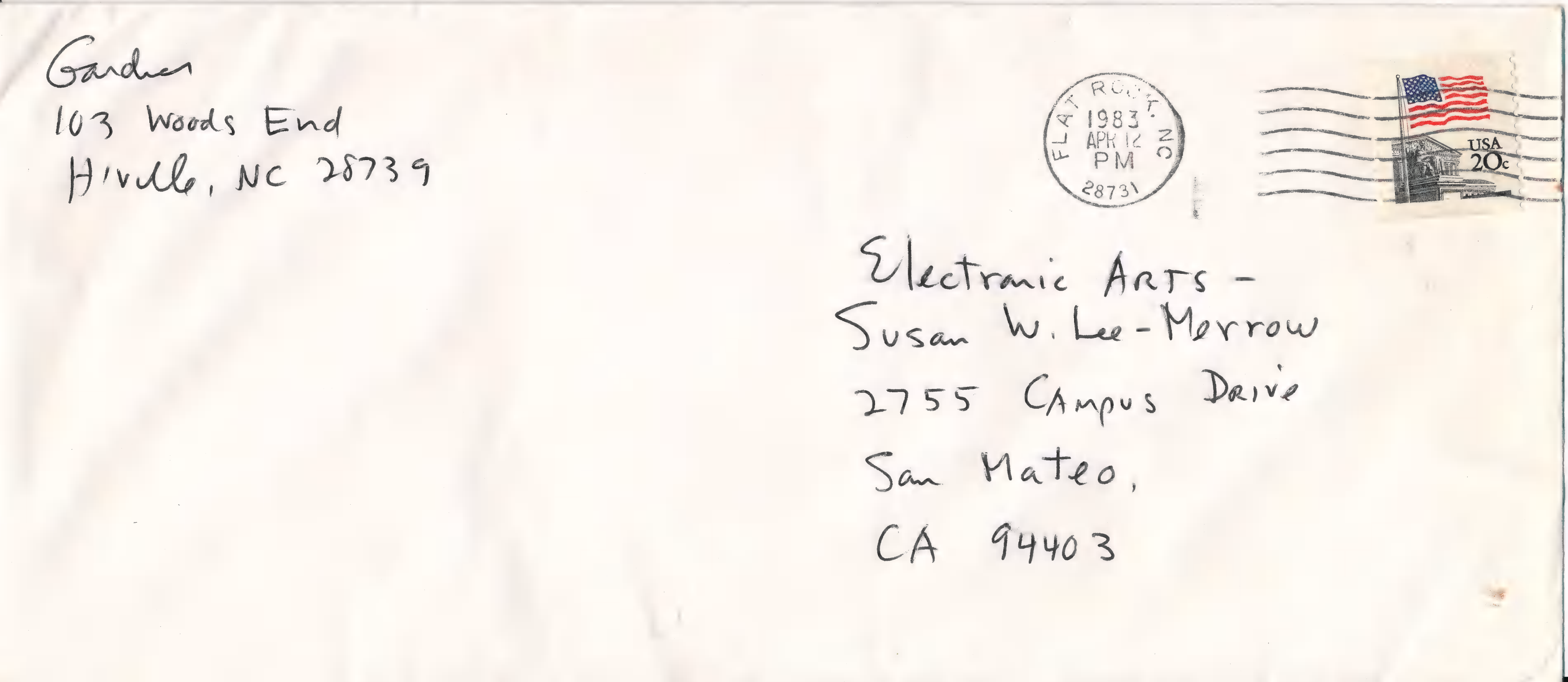
Not
Fox & Cushadon

Attempted-Not Known
No Such Number
Vacant
Refused
No Mail Receptacle
Temporarily Away
Route # 313
Initials SE

ADDRESS
UNKNOWN
RETURN TO WRITER

SAN MATEO, CA 944
PM
APR 7
1983

Susan W. Lee-Merrow
2755 Campus Dr.
San Mateo,
CA
94403.



Gardner
103 Woods End
H'ville, NC 28739

FLAT ROCK, NC
1983
APR 12
PM
28731

USA 20c

Electronic Arts -
Susan W. Lee-Merrow
2755 Campus Drive
San Mateo,
CA 94403

M. G. (pres)

Woods End, Inc.
103 Woods End Drive
Hendersonville, North Carolina 28739
Telephone (704) 693-3810

Dear Patricia Marriott: 28 Jan 83

As you can see, I'm now stuck away in the mountains of western NC, and your letter of Jan 19 has just reached me.

Glad to see a Worms program being made available. I am sure you must have considered Conway's Life, about which I did two columns in 1970 and 1971. I am told that William Fredkin, the AI expert, has the world's fastest Life chip. His program even allows you to vary the transition rules of Conway's famous game. I am currently doing a new book of my SA columns, to be published by W. H. Freeman this year, that will end with my two columns on Life, plus a third one I'm doing specially for the book that will update recent developments. I have been amazed for years that no one seems to have provided Life software for computer buffs. If you can contact William Gosper, who works I think for Xerox in Stanford (home phone 415-948-2149), maybe you could visit him and see the 17-minute film on Life he prepared when he was in Minsky's department at MIT. Its much more fantastic to watch on a scope than the crawling worms!

best,

Martin Gardner

RECEIVED
FEB 01 1983

MARTIN [illegible]DNER
103 WOODS END DR.
HENDERSONVILLE,
N.C. 28739

...LLE, NC 28...
PM
JAN

USA
20c

Patricia Marriott
Elect. Arts
500 Airport Blvd.
Burlingame, CA 94010

ELECTRONIC ARTS
500 AIRPORT BOULEVARD
SUITE 440
BURLINGAME, CALIFORNIA
94010
(415) 340-0454

Mr. Martin Gardner
℅ Scientific American
415 Madison Avenue
New York, N. Y. 10017

January 19, 1983

Dear Mr. Gardner:

I am a producer in a new software publishing company, Electronic Arts. We believe that the home computer potential has yet to be realized. Our goal is to fulfill the promise of home computing by providing high quality software for entertainment, education, and home information management.

I recently acquired a program based on your Mathematical Games in the November 1973 Scientific American. The subject is worms—moving points programmed to generate a pattern of lines. This program is one of the most unique mathematical games I have ever seen on a microcomputer. As the worms move, the computer generates music related to the worms' pattern on the screen. Thus, each instance of game play results in a unique graphic and musical pattern.

I would like to discuss "Worms" with you and hopefully give you the opportunity to see it! I would also like to mention your name, and perhaps some comments you may have on the game, when we publish "Worms" this spring.

Please call me at (415) 340-0454, or write to me at Electronic Arts. In your article you wrote, "if one has access to a computer with a display screen, one can ... enjoy the spectacle of watching a path of light grow on the screen." I believe you would enjoy seeing and playing this mathematical game.

Sincerely,

Pat Marriott

Patricia Marriott
Producer

Sci Am. (212) 754-0550 415 Madison Ave NY 10017

MATHEMATICAL GAMES

Fantastic patterns traced by programmed "worms"

by Martin Gardner

Fashionable methods of teaching mathematics to children come and go. (At the moment the modern-math fad is mostly going.) One of these days mathematics teachers will discover what John Dewey tried to tell them 75 years ago: Children learn best by doing something they enjoy. With this in mind, Seymour A. Papert, a former assistant to Jean Piaget who is now working in the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology, has designed a variety of animal robots that can be controlled by a desk computer. One of them is a "turtle" with a pen on its underside. Suitably programmed, the turtle draws geometric figures by crawling across large sheets of paper on the floor.

Defining geometric figures as paths generated by a moving point is an ancient idea. Take a square. Instead of calling it a four-sided polygon with equal sides and angles, call it the path traced by a worm crawling over a plane according to the following rule: Go straight for distance k, turn 90 degrees left and repeat until the path returns to its origin.

The idealized worm (moving point) can obviously be programmed to generate any pattern of lines. A challenging recreational task now presents itself. What kinds of program, with extremely simple rules, give interesting or beautiful patterns? A good way to simplify rules is to restrict the worm to paths along a regular lattice. This enables one to experiment with rules by drawing paths on square or isometric graph paper. Better still, if one has access to a computer with a display screen, one can write simple programs, then enjoy the spectacle of watching a path of light grow on the screen.

Frank C. Odds, a British biochemist, recently proposed a class of rules for generating patterns that he named spirolaterals. The worm crawls a distance of one unit, turns, crawls two units, turns, crawls three units and so on, traversing distances in counting order until the length of a path segment reaches a specified integer, n, when the procedure starts over. The turning angle is always the same, but the direction of turn may be right or left according to the worm's program. The number n, which is the number of segments in the counting series and also the number of turns before the series repeats, is called the order of the spirolateral.

Two examples will make this clear. If the order is 1, the angle is 90 degrees and all turns are in the same direction, the spirolateral is a square. If the order is 7, the angle is 90 and all turns are the same, the spirolateral has a pleasing closed pattern. The top illustration on page 118 shows all the right-angle spirolaterals of orders 1 through 8, when all turns are the same. It is easy to see why Odds chose the name spirolateral: "lateral" for flat surface and "spiro" for the square spirals that generate the figures.

Note that spirolaterals of orders 4 and 8 do not close (return to the origin). As Odds puts it, they meander jerkily to infinity. Indeed, it is not hard to prove that there is no closure for orders in the series 4, 8, 12, 16..., that two repetitions of the square spiral will close the figure (producing twofold symmetry) if the order is in the series 2, 6, 10, 14..., and that all other orders close after four repetitions and have fourfold symmetry.

There is no reason why all turns must be the same way, but when they vary, the situation becomes complicated enough to call for a compact notation. Odds suggests writing the angle as a subscript of the order number and using a superscript on the left or the right to show which turns are left and which right. For example, ${}^{1,7}9_{90}$ defines the spirolateral generated by nine turns of 90 degrees, with left turns preceding segments 1 and 7 and (by implication) all the other segments turning right. The same spirolateral could be defined by putting a superscript on the right to indicate right turns, ${9_{90}}^{2,3,4,5,6,8,9}$, but clearly the first notation is preferable.

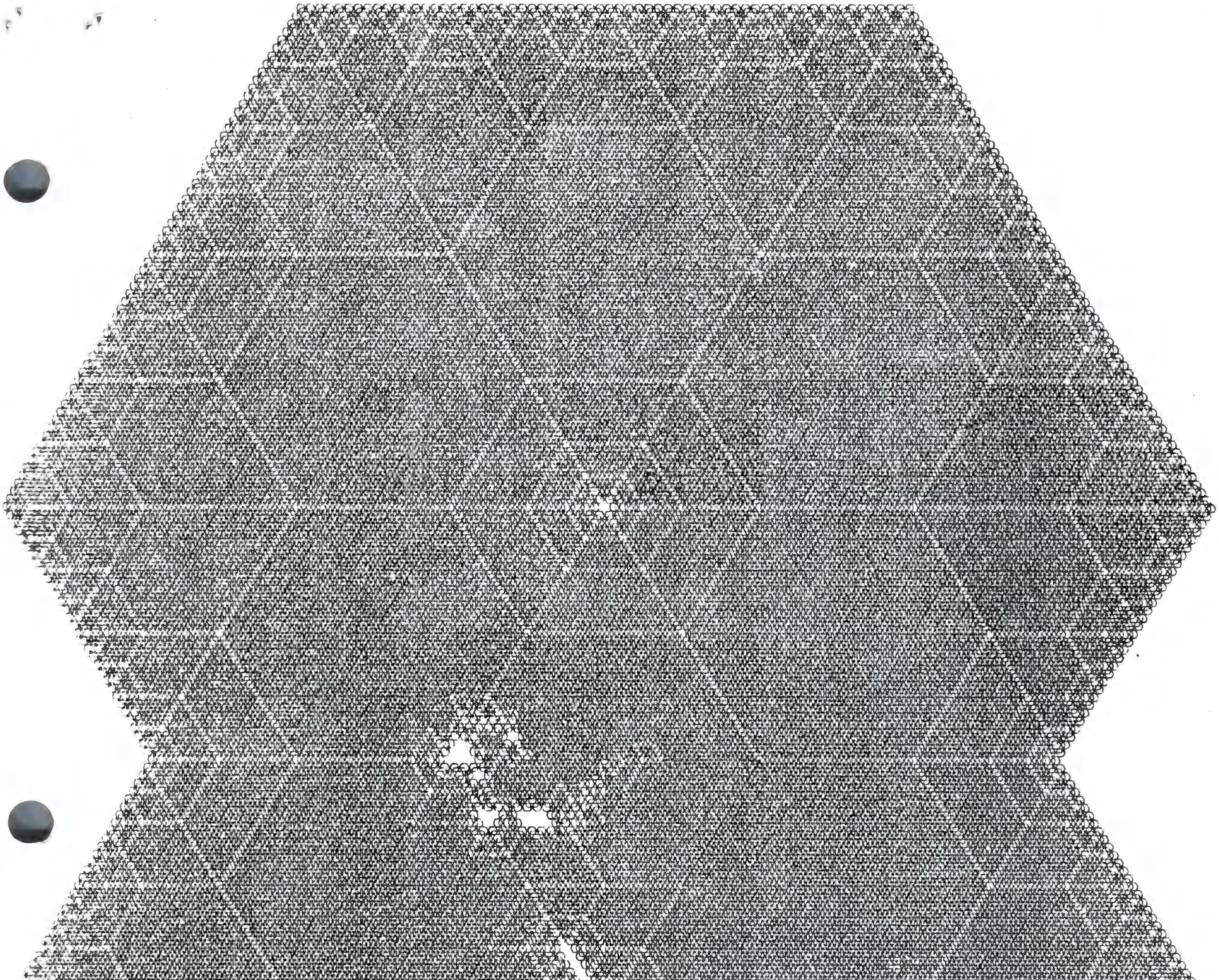
The middle illustration on page 118 shows two right-angle spirolaterals with mixed left and right turns. The bottom illustration shows spirolaterals with angles of 36, 45 and 60 degrees. Spirolaterals with 60-degree turns can be easily drawn on isometric paper. Those with 45-degree angles are not hard to construct with a ruler and a draftsman's triangle, but those with other angles take more doing. Any angle that is an exact divisor of 180 degrees will generate a spirolateral.

Not much is known about spirolaterals. Sometimes two different rules produce spirolaterals that are mirror images (for example ${}^{1,2,3}7_{90}$ and ${}^{5,6,7}7_{90}$), but no one knows how to predict this without drawing the figures. Nor is it known how to tell from looking at a spirolateral's formula, except in special cases, whether or not the figure will close and, if it does, how many repetitions are needed to close it.

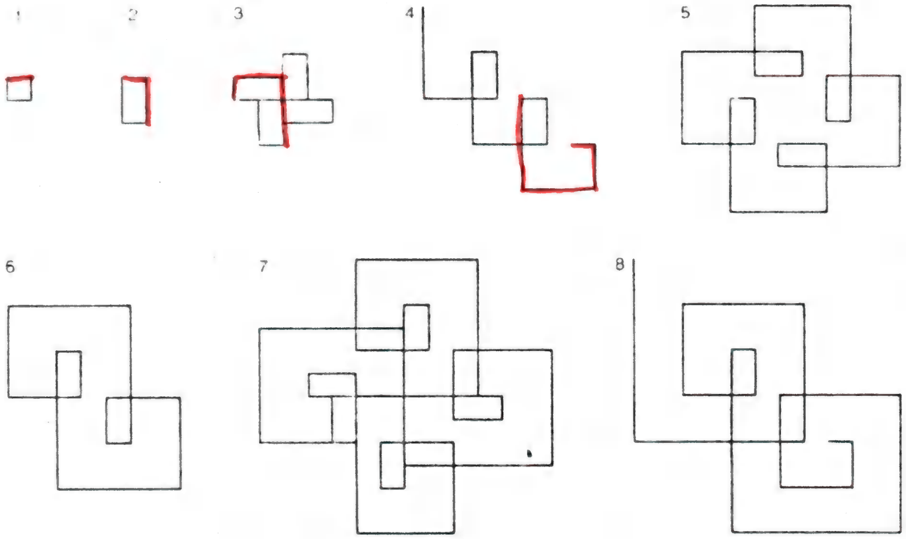
A few years ago John Horton Conway of the University of Cambridge suggested a new approach to worm paths. Instead of viewing the worm as an explorer, view it as an eater. Food is confined to the lines of an arbitrarily large grid. The worm hatches from an egg at one node and then starts to crawl along grid lines, eating as it goes. A fixed set of rules determines the worm's decision at each node. It is assumed that the worm never traverses a segment already eaten. No segment lengths need to be specified, only the direction of turn, the direction depending solely on the state (eaten or uneaten) of all segments meeting at that node.

"Paterson's Worm," by Michael Beeler, a memorandum issued in June of this year by M.I.T.'s Artificial Intelligence Laboratory, deals entirely with such worm tracks. What follows, either directly quoted or paraphrased, is taken from the memorandum with the permission of Beeler and Marvin L. Minsky, who heads the laboratory. (Readers can obtain the document by sending 80 cents to Room 912, 545 Main Street, Cambridge, Mass. 02139.)

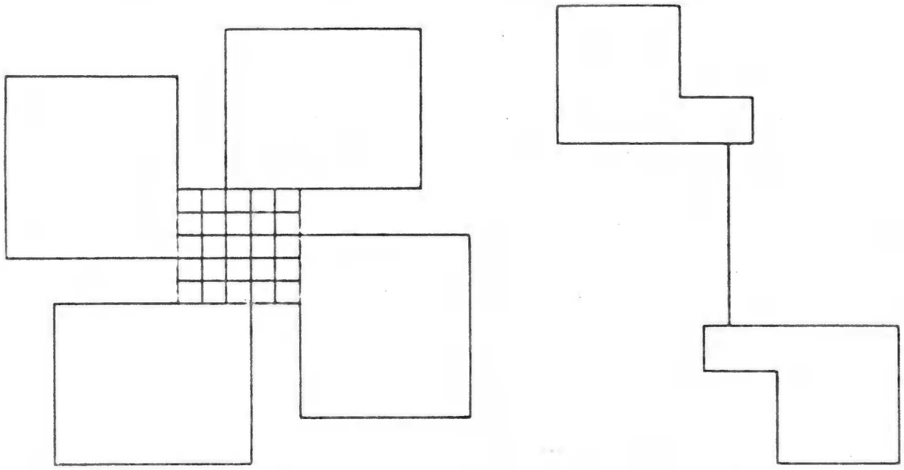
"Certain prehistoric worms fed on sediment in the mud at the bottom of ponds," Beeler's memorandum begins. "For efficiency, they would not retrace paths which had already been traveled, since little food was left there. Yet food probably occurred in patches, so it was desirable to stay near previous trails. Worms had innate 'rules' regarding how close to 'eaten paths' to stay, how far to



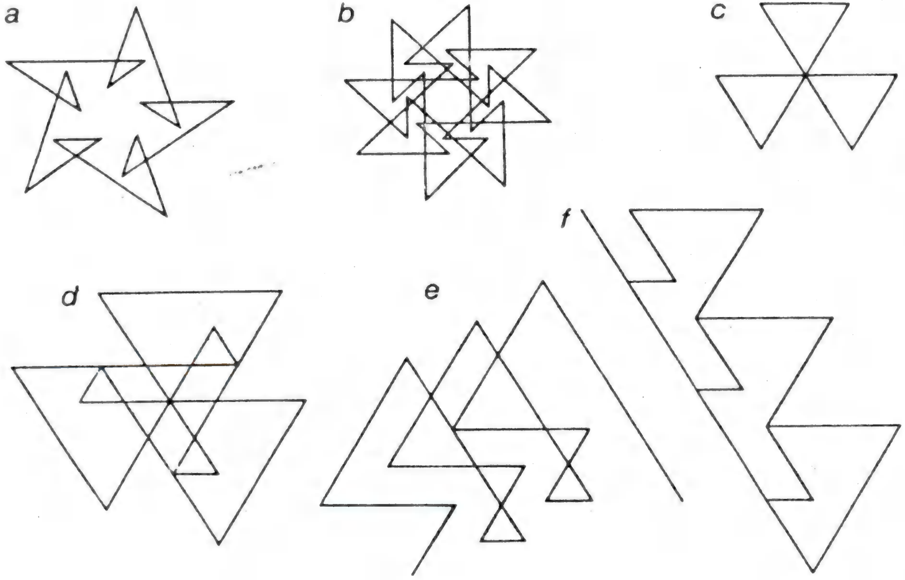
Longest known finite path generated by a simple isometric worm



Spirolaterals of orders 1 through 8 with all turns in same direction



Ninety-degree spirolaterals with mixed turns: 9_{90}^{6} (left) *and* $8_{90}^{1,4,8}$ (right)



Spirolaterals (a) 3_{36}, (b) ${}^{1}3_{45}$, (c) 2_{60}, (d) ${}^{1,2}5_{60}$, (e) ${}^{3}5_{60}$ *and* (f) ${}^{2}5_{60}$

go before turning around, how sharp a turn to make, etc. These rules varied from species to species, and paleontologists can trace the development of species and determine the similarity of different species by comparing fossil records of worm tracks. (See *Science* magazine, 21 November 1969, for further details and a discussion of computer simulation of natural worm tracks.)

"Early in 1971, Michael Paterson [a computer scientist at the University of Warwick] mentioned to me a mathematical idealization of the prehistoric worm. He and John Conway had been interested in a worm constrained to eat food only along the grid lines of graph paper. . . .

"If a worm, arriving at a node with no segments eaten (except of course the one it just ate), should find in its rules, 'For this distribution, go straight,' then the worm will go straight forever. Since this is neither interesting to us, nor very useful to a real worm, which would quickly reach the edge of its food patch, we discard it. We require that, upon discovering a virgin node, all sets of rules must say to turn. To avoid mirror-image duplication, we require that the turn be to the worm's right (clockwise as seen from above)."

Consider what Beeler calls a "simple quadrille worm": a worm crawling on a square grid and turning right at each node. What happens after it traces a square? It cannot turn right again because this would take it along an eaten segment. It has only two choices. If it is programmed to turn left if, and only if, it cannot turn right, it will trace two squares [*see "a" in top illustration on opposite page*]. No uneaten segments are available, and so the worm dies. If it is programmed to go straight when it cannot go right, and left when it cannot go straight or right, it will trace five squares before it expires at its origin ["*b*"]. These two fossil tracks exhaust the variety of species of simple quadrille worms.

To avoid a worm's early demise, Conway proposed that quadrille worms have the ability to "look ahead" and see the distribution of eaten and uneaten segments at each adjacent node. For example, a worm could be programmed to turn right if, and only if, it senses that this will take it to a node where four uneaten segments meet. Otherwise it goes straight. The result is a simple square spiral. If the rule is to turn left whenever a right turn leads to a node with an eaten segment, the path is a more interesting spiral [*see middle illustration* at right]. It is easy, of course, to produce more elaborate paths by complicating the rules.

The rules may allow anything. What happens when rules allow look-ahead worms to hop? What happens when barriers are suitably placed or when the [illegible] is bounded on all sides? What happens when two or more worms of the same or different species interact? What happens when a newly hatched worm crawls a short distance along a defined path (such as a straight line of three units) before its repetitive behavior starts? How about two armies of worms crawling toward each other, each army obeying a different program? Are there possibilities here for competitive games? Are there interesting paths or patterns in three dimensions and higher dimensions?

Beeler avoids such difficult questions by confining his attention to what I shall call "simple isometric worms": worms that feed along an isometric grid of unit equilateral triangles. They are simple worms because they do not look ahead. With the isometric grid, however, six segments (not four) meet at each node. It seems to be a trivial difference, but as Beeler makes clear the possibilities for variant rules allow the definition of no fewer than 1,296 species.

All simple worms, quadrille or isometric, obey three general rules:

[illegible] If no segments have been eaten at a node (except the segment just traveled), the worm turns right.

2. If all segments at the node have been eaten, the worm dies.

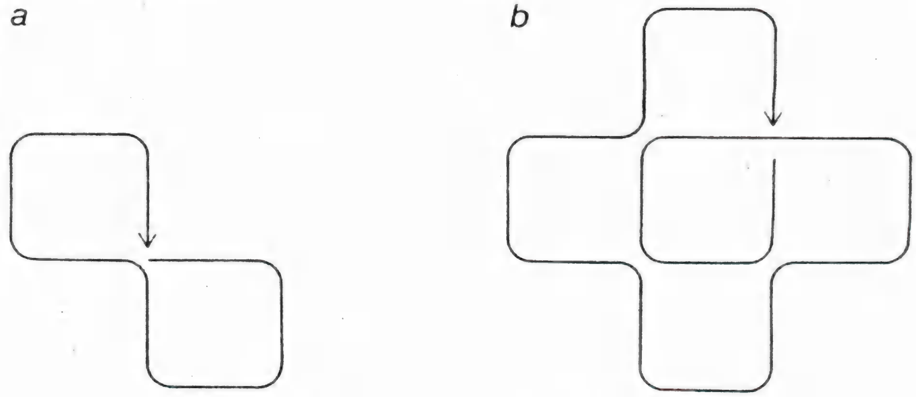
3. If only one segment at the node is uneaten, the worm takes it.

As we have seen, a simple quadrille worm, following the above rules, encounters only one "field" in which it must choose. Since it has only two choices, we can define only two species. The isometric grid, in contrast, offers a simple worm four major fields (one consisting of four subfields) in which decisions must be made. It is the behavior of these simple isometric worms that Paterson became interested in, and Beeler's memorandum also is primarily concerned with their behavior.

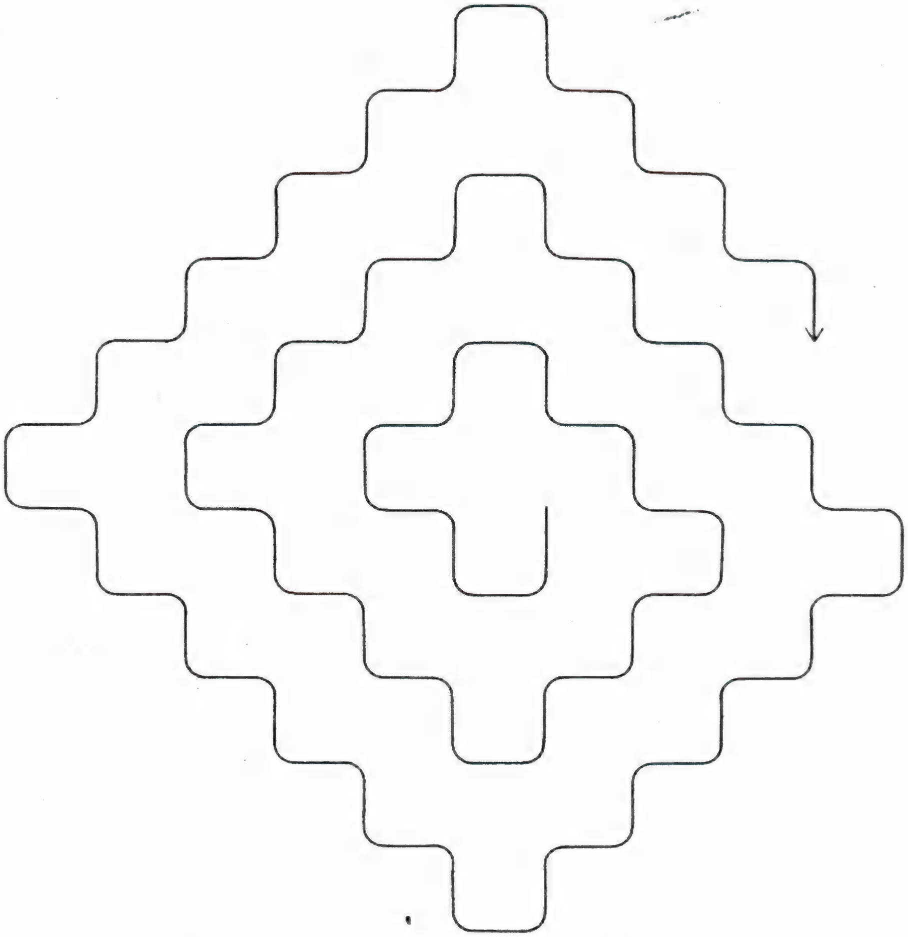
The four major fields, and all their choices, are shown in the illustration on the next page. The black lines indicate uneaten segments, the light colored lines are eaten segments of the worm's path and the colored arrows show how the worm approaches and leaves a node.

The four fields are:

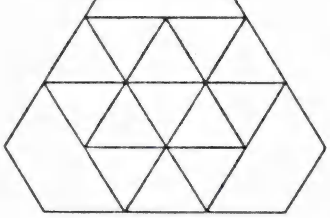
[illegible] The worm approaches a node with no eaten segments other than the one it has just traversed. Its right turn can be



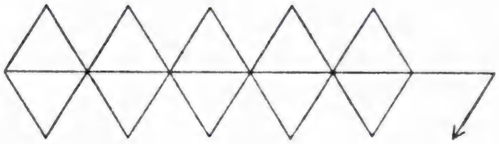
Fossil tracks of the only two species of simple quadrille worms



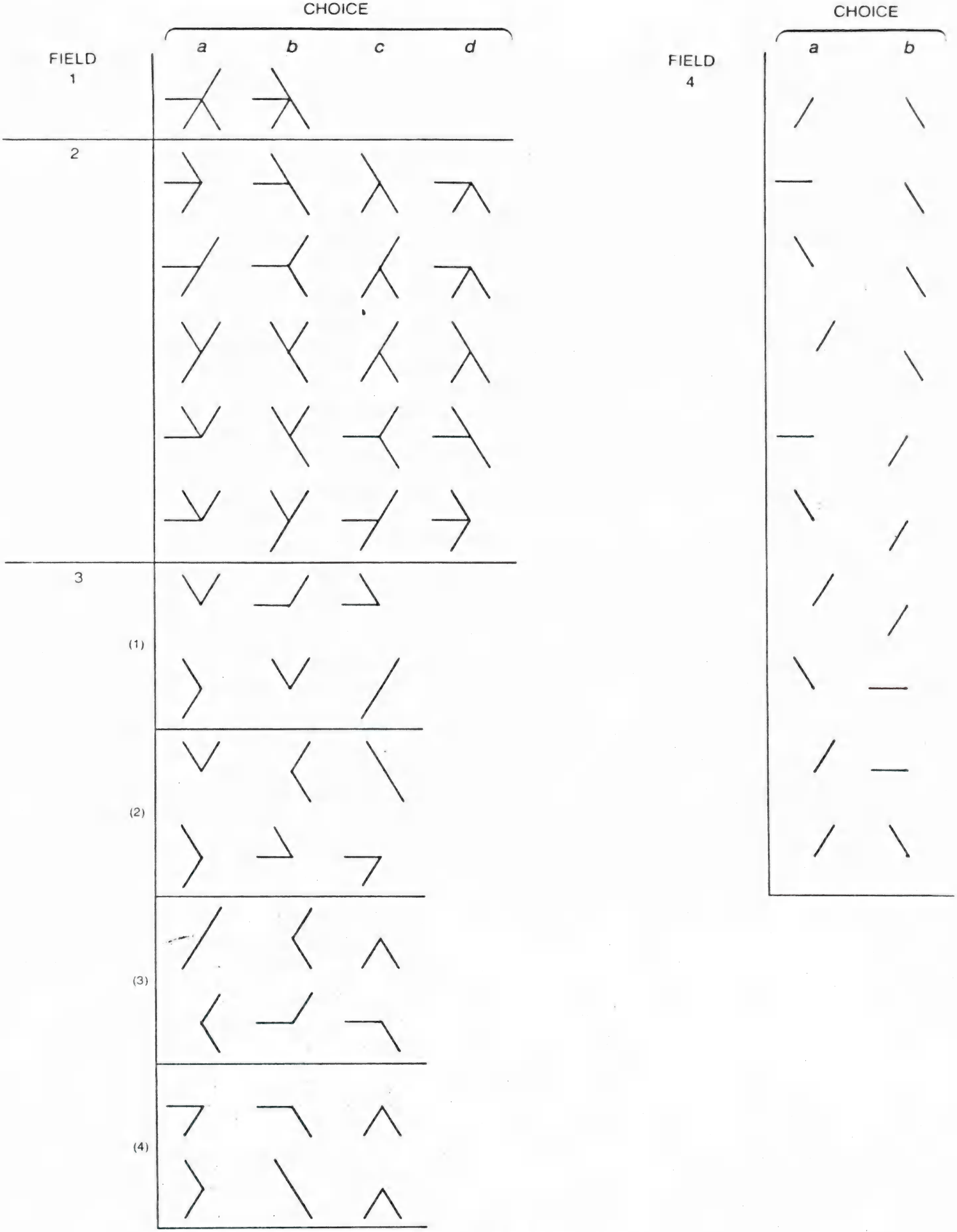
Infinite path of a "look ahead" quadrille worm that turns left when it cannot turn right



Path traced by 14 simple isometric worms



Zipper path of 54 sharp worms



Fields of choice for simple isometric worms either “gentle” (120 degrees) or “sharp” (60 degrees). Number of choices: two.

2. The worm meets one eaten segment as it returns for the first time to its origin. As the chart shows, it can approach this node along any of five different segments. For each approach there is a choice of leaving by one of four uneaten segments. Number of choices: four.

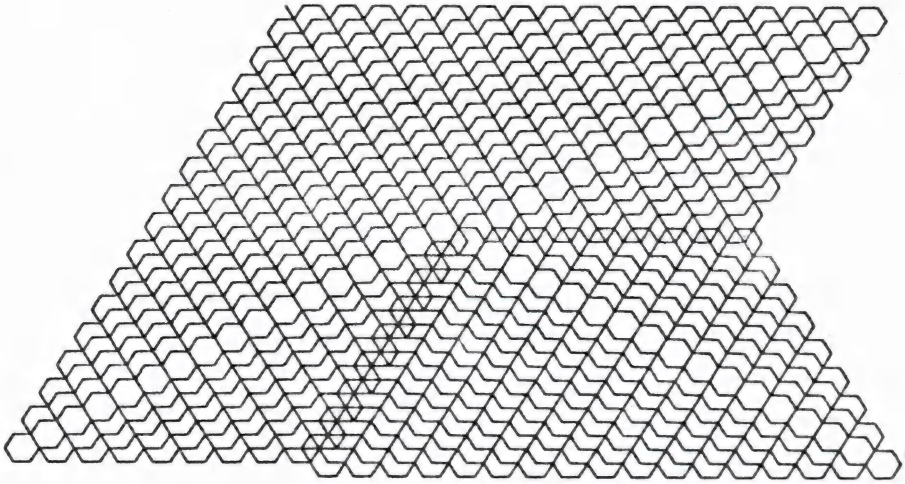
3. The worm meets two eaten segments as it returns to a point along its path. The node will be the vertex of either a sharp turn or a gentle one. In either case the approach can be made in four ways. For each way the worm can leave by one of three uneaten segments. Number of choices: 3 × 3 × 3 × 3 = 81.

4. The worm meets three eaten segments as it returns for the second time to its origin. This can happen in 10 ways, but each requires only a choice between two ways of leaving the node. Number of choices: two.

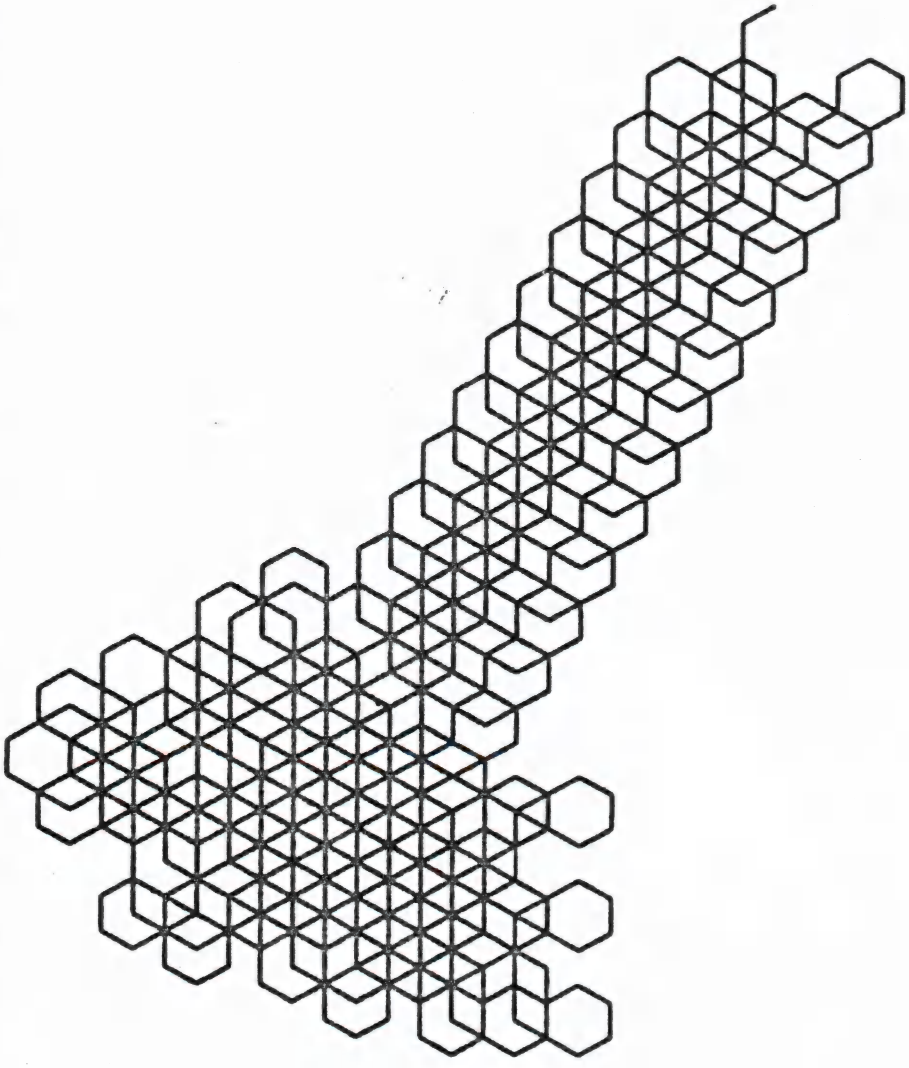
Meeting four or five eaten segments does not offer the worm a choice. In the first case it must take the only remaining uneaten segment. In the second case it has no segment to take, and so it dies. Thus there are 2 × 4 × 81 × 2 = 1,296 sets of rules, each defining a distinct species of simple isometric worm.

Beeler uses the pattern shown in the bottom illustration at the left on page 119 to explain how the rules work. The notation adopted here (Beeler uses a more compact notation based on octal and binary digits) identifies this as a fossil path generated by a $1_a2_b3_{acac}4_b$ worm. The formula tells us that when the worm faces a choice in Field 1, it follows Rule *a*. (It makes a gentle, not a sharp, turn.) For Field 2 it chooses *b*. The four subscripts after 3 refer to the four subfields of Field 3, where choices *a*, *c*, *a*, *c* are in the worm's feeding program. Finally, for Field 4 the worm selects *b*. As Beeler suggests, the reader might pause at this point to see if he can trace the path on a sheet of isometric paper, following the seven rules given by the formula. (It is necessary at each node to rotate the paper until the pattern at that node corresponds to the proper diagram on the chart.)

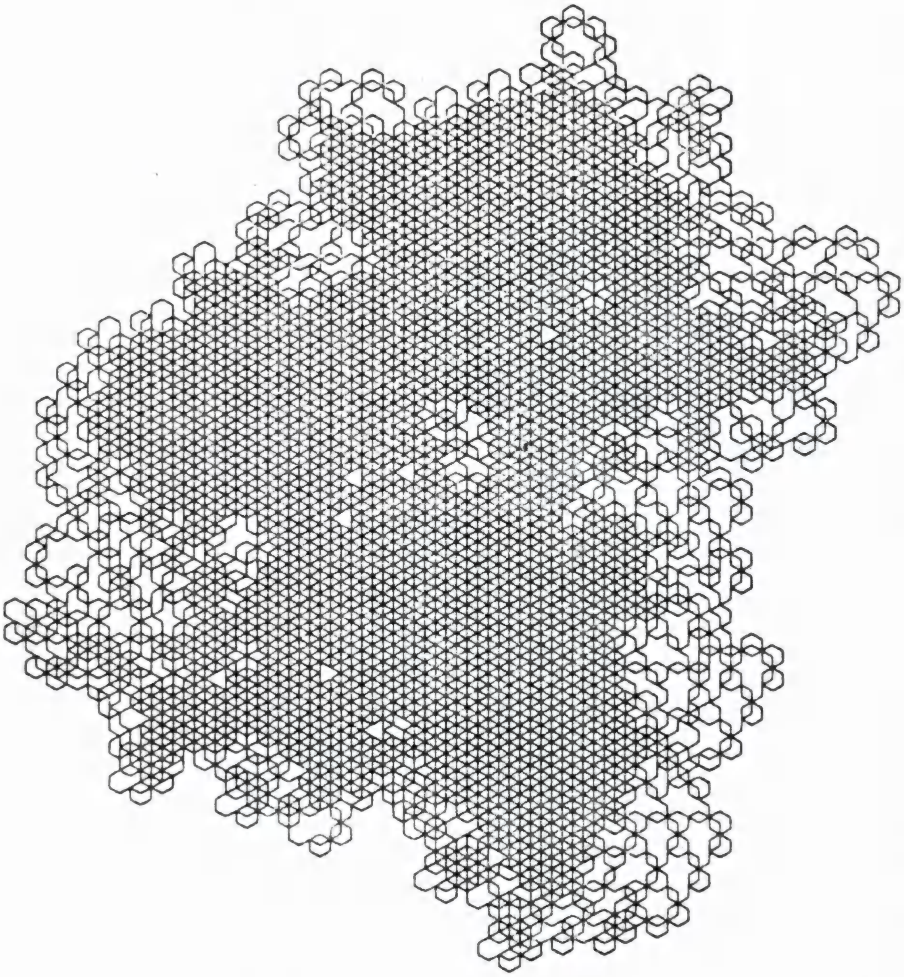
The path just described is generated by 14 worms. Some trace it exactly the same way. For example, the choice presented by the first subfield of Field 3 never arises, therefore it does not matter which of the three choices, *a*, *b* or *c*, is in the program. We indicate this in the formula by putting the three choices inside parentheses: $1_a2_b3_{(abc)cac}4_b$. The formula now describes three different



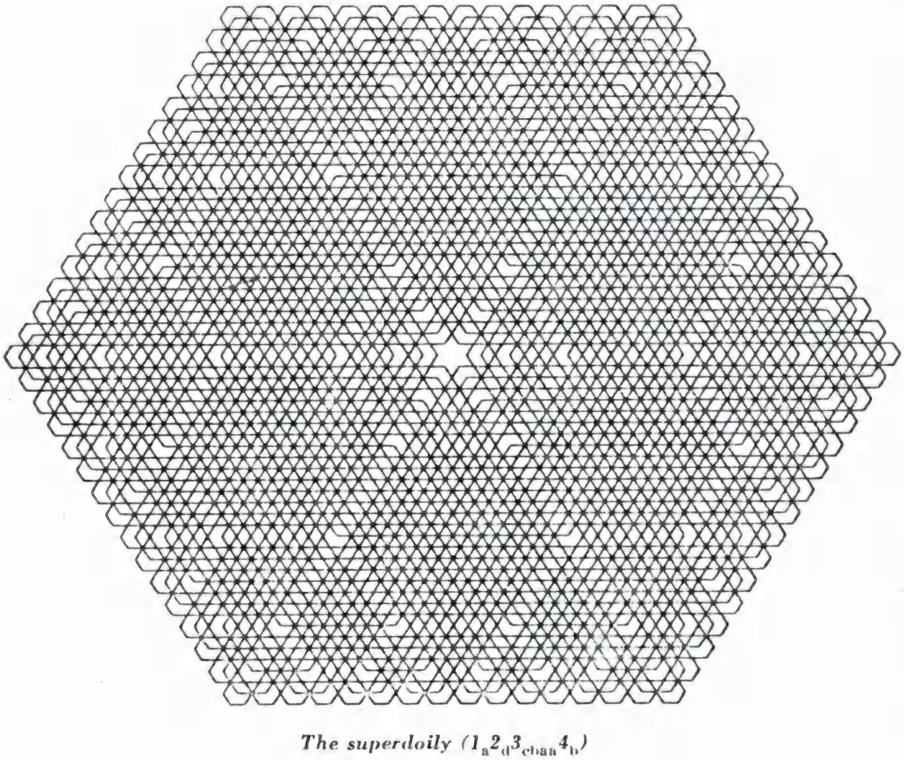
Three-pointed spiral by a $1_a2_c3_{acba}4_a$ worm



Shoot grower by a $1_a2_d3_{caaa}4_b$ worm



Cloud path generated by a gentle worm $1_a2_{abaa}3_c4$



The superdoily ($1_a2_d3_{cbaa}4_b$)

worms. Each draws the same path the same way. Other worms draw the same path different ways.

Beeler's computer program explored the behavior of all 1,296 species of simple isometric worm. Results show that 209 species generate paths that are unique, in the sense that no other species generates them. Forty-six paths are each characteristic of two species, and 44 paths belong to more than two. Thus there are 299 distinct paths all together.

The simplest of the 299 paths is the radioactivity symbol [*see "c" in bottom illustration on page 118*]. It has the smallest length (nine units) and also the fewest number of nodes (seven). It is the path traced by the largest number of worms, no less than 162 species. Using parentheses to show alternate choices, one formula defines all 162:

$$1_b2_{(ac)}3_{(abc)(abc)(abc)(abc)}4$$

The formula shows at once that it describes 2 × 3 × 3 × 3 × 3 = 162 sets of rules. If the reader will test this formula on isometric paper, he will find that, regardless of the choice he makes inside each pair of parentheses, the radioactivity symbol will result.

Is there a longest path? No, because many paths never return to their origin a third time and are therefore infinite. A trivial infinite path of a type called the zipper is produced by 54 sharp worms with the formula $1_b2_d3_{(abc)(abc)b}4_{(ab)}$ [*see bottom illustration at right on page 119*].

Other worms spiral forever around their origin. The spiral shapes vary enormously: hexagons, diamonds, triangles, stars and asymmetric shapes such as the one shown in the top illustration on the preceding page. "Shoot growers," another class of infinite paths, start in a conventional way, and then the worm falls into a curious spiraling action that keeps repeating itself with regular displacements, creating a rod that shoots off to infinity [*see bottom illustration on preceding page*].

The longest of the known finite paths is shown in the illustration on page 117. The path is generated by a $1_a2_d3_{cbac}4_b$ worm and has 220,142 units. Note the crystalline regularity of the border and the lines that crisscross near the border, and how strikingly this symmetry contrasts with the dishevelment near the center. The point of origin, not identifiable, is at a spot just to the left of the center of the pattern.

One speaks of "known" finite paths because about a dozen worms have paths so long that no one yet knows whether they are finite or infinite. The path generated by $1_b2_a3_{bcaa}4_b$, for instance, was tracked by Beeler to a length of 10 million units without revealing whether the worm dies or goes on gorging itself forever.

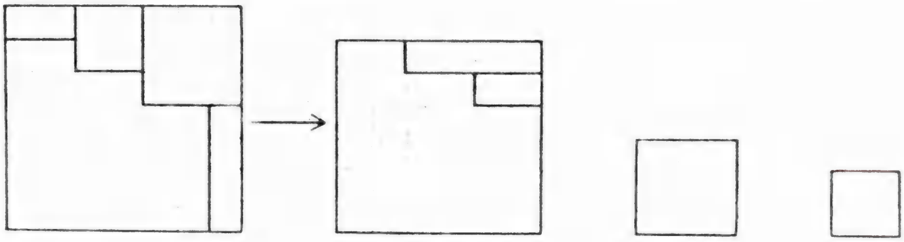
Some path patterns have outlines as unstructured as a cloud [*see top illustration on opposite page*]. Others, such as the "superdoily," display the rigid sixfold symmetry of a snow crystal [*see bottom illustration on opposite page*]. Note the six-pointed star in the center. Occasionally one such star or more appear as randomly situated white spots inside a gray plenum of closely packed unit triangles. Sometimes the stars are separated; sometimes they are partially merged to form binary or ternary systems. The situation is not unlike that of the physical world, in which low-order mathematical laws generate a rich assortment of both patterned and unpatterned objects, as well as objects in which order and disorder, symmetry and asymmetry are crazily mixed.

Beeler's work clearly has just scratched the surface of isometric-worm pathology. His program investigated only the simplest genus, with no attempt to complicate the genetic rules or to find out what happens when simple worms interact with barriers, boundaries or other worms of the same or different species.

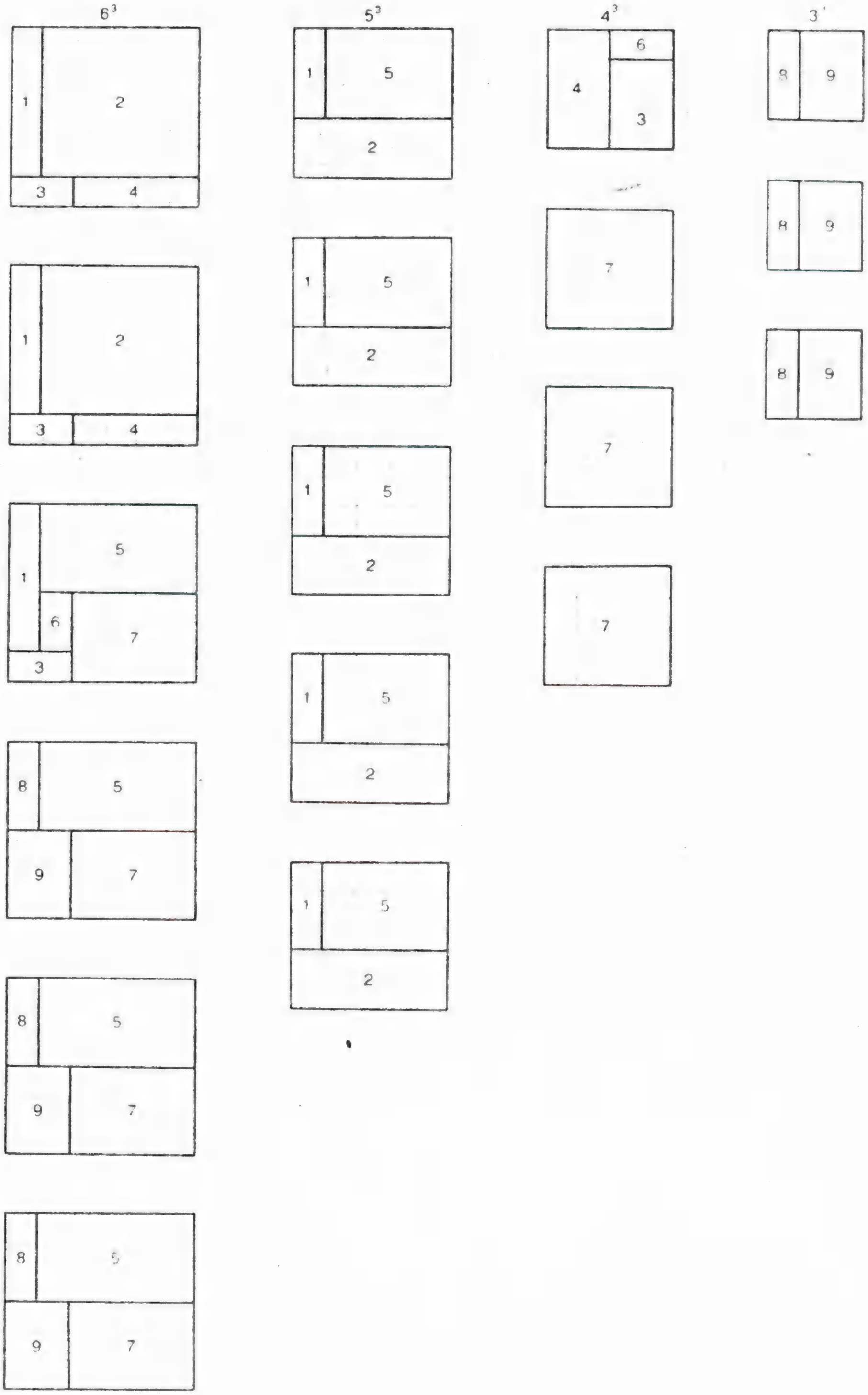
On August 13 *Newsweek* ran a story about Papert's turtle. A girl who had ranked near the bottom of her mathematics class was programming the turtle to draw a certain design. "That math must be fun," said a passing visitor. "There ain't *nothin'* fun in math," the girl replied. She had no idea she was doing math, reported *Newsweek*, and Papert saw no reason to tell her.

Last month's problem was to improve on a dissection of the 7-square into six pieces that form squares of sides 2, 3 and 6. A five-piece solution is shown in the top illustration at right. This dissection has the minimal cutting length of 16 units.

The bottom illustration at the right shows in six cross sections how Thomas H. O'Beirne sliced a 6-cube into nine rectangular blocks (the minimum) that can be reassembled to make separate cubes of sides 3, 4 and 5. No two "bricks" in this remarkable dissection are alike. O'Beirne has shown that a dissection in eight blocks is not possible, but it is not yet known whether or not the nine-piece dissection is unique.



Five-piece dissection for $7^2 = 6^2 + 3^2 + 2^2$



Nine-block dissection for $6^3 = 5^3 + 4^3 + 3^3$

Gordie

Not trade secret
Not copyright-able
Probably not patentable if it
hasn't been done yet!

Martin Gardner — Sci American
John Conway - Math

11/9 - David - Easy to convert for Commodore 64.

Gordie 1/19/83

Guy at MIT may have copyrighted his game

Downside — if Gardner is used as endorsement
people may assume we had access to
Beeler's. — but David did not see Beeler's.
However, it's hard to prove a negative

Best to use Gardner as general endorsement
vs specific to worms.

ELECTRONIC ARTS
500 AIRPORT BOULEVARD
SUITE 440
BURLINGAME, CALIFORNIA
94010
(415) 340-0454

2/22/83

Dear David,

Per our Software Development Agreements for Sumo Worms, I am formally requesting that you develop the Commodore 64 Derivative Work for us. Under the terms of the contract, you should respond to me in writing within 30 days.

It may seem a bit early to request this port, but we would like to ship the derivative as close on the heels of the original as possible. Therefore, we should initiate discussions regarding specifications and delivery schedules. From our previous conversations I am hopeful and confident that you will be interested in doing the derivatives.

I look forward to working with you in taking Worms to this new market.

Sincerely,

Pat

Pat Marriott, Producer

ELECTRONIC ARTS
500 AIRPORT BOULEVARD
SUITE 440
BURLINGAME, CALIFORNIA
94010
(415) 340-0454

This is to acknowledge receipt of the prototype version of Sumo Worms for the Atari 800. The code was electronically transmitted by the author, David Maynard, 424 Oak Grove, Menlo Park, CA and received by the producer, Pat Marriott, at Electronic Arts January 12, 1983.

Patricia Marriott 1/12/83

ELECTRONIC ARTS
2755 CAMPUS DRIVE
SAN MATEO, CA 94403
TEL 415 571-7171

This is to officially acknowledge delivery of WORMS? electronically on April 19, 1983. Source Code and Object Code were delivered via modem.

Susan W. Lee-Merrow

Susan W. Lee-Merrow
Producer

ELECTRONIC ARTS
2755 CAMPUS DRIVE
SAN MATEO, CA 94403
TEL 415 571-7171

This is to officially acknowledge acceptance of WORMS? as a finished product to be published by Electronics. Acceptance has taken place May 11, 1983.

Susan W. Lee-Merrow

Susan W. Lee-Merrow
Producer

WORMS? CONTEST

- Train a worm and send it in for a contest
- There are 64 possible decision-making points
  - WILD predetermines ~~the~~ all 64 decisions before play begins
  - AUTO makes ea. decision automatically as it is faced
  - NEW
    - keep playing against WILD until ~~you~~ make no more new moves
    - can have chart of all 64 & check each off as it is encountered
    - if worm must stop, you loose
    - can you mix worms?
    - can you send in any worm (WILD, AUTO, NEW)?

- each entrant gets a T-shirt? (entrant must send disk)
- winner?
- how many worms per entrant?
- who administers (and judges)?
- hierarchy for winning? play 2, 3, or 4?

3·14·83

1. Contest considerations

2. Keyboard input — space bar moves worm/any other key is anchor

3. worm editor: graphically or text notation
   2 $3/disk
   paper + printer interface
   or step through worm

people send in disks
get t-shirt

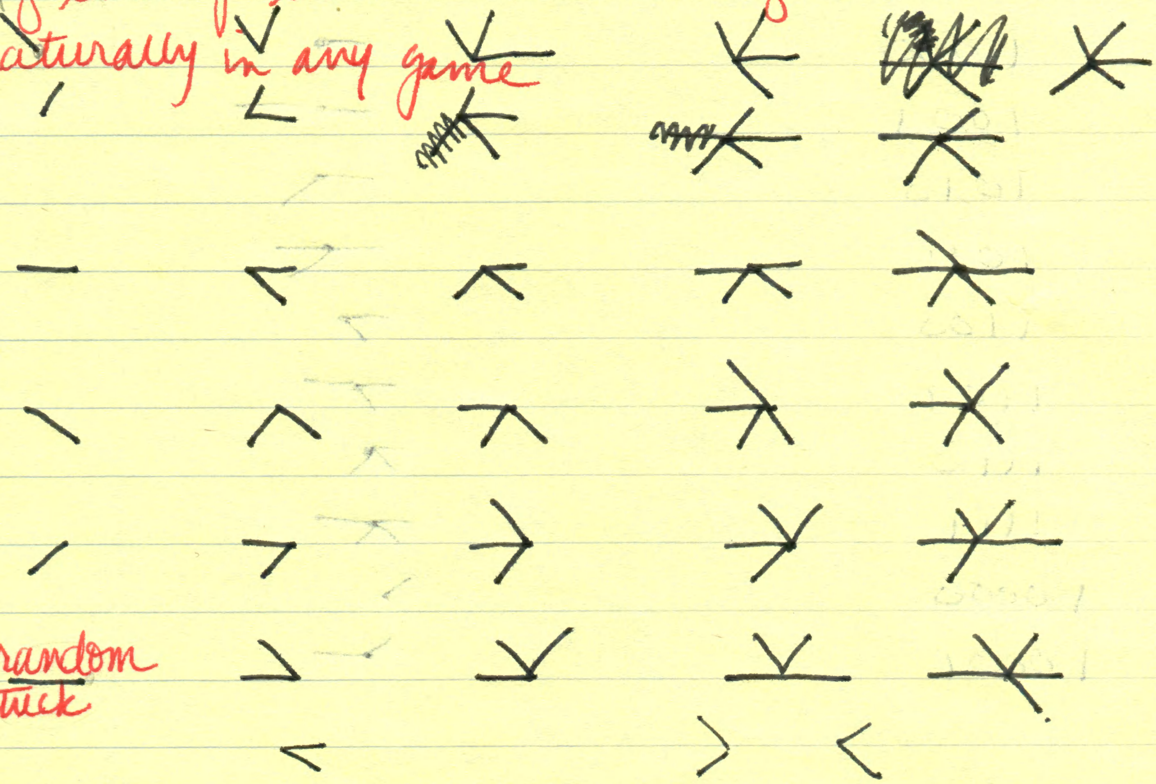
prizes
stickers on pkg.

Manual

— give all 64 permutation

Worm Contest —

must have worm programmed to respond to all possible permutations (64) of situations: and there's no way to do it because of lack of editor + not having all situations occur naturally in any game



64 permutations

-editor

-programmed for random move when stuck

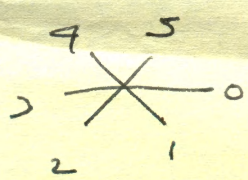
- sent worms over phone? (need DOS files) = just for worms

– in package — what about when deadline is over + pkg is still on shelf

problems

~~difficult to reach all 64 positions, unless you have programmed them all in~~

– not clear hierarchical structure of winning A > B > C > A etc



000000
000001
10
11
100
101
110
111
1000
1001
1010
1011
1100
1101
1110
1111
10000
10001